

Chương 6

TÍCH PHÂN SỐ

Chương này trình bày một số thuật toán và chương trình tính tích phân xác định của các hàm giải tích bằng phương pháp số.

Bài số 68 Tính $\int_a^b f(x)dx$ bằng phương pháp Simpson.

Bài giải Cho hàm $y = f(x)$ liên tục trong khoảng $[a,b]$, tích phân xác định của hàm $y = f(x)$ trong khoảng $[a,b]$ được tính gần đúng bằng phương pháp Simpson như sau :

$$\int_a^b f(x)dx = \frac{h}{3}[f(a)+f(b) + 4\sum_{i=1,3}^{m-1} f(a+ih) + 2\sum_{i=2,4}^{m-2} f(a+ih)] + R_m(f)$$

trong đó phần dư $R_m(f) = -\frac{h^4}{180}(b-a)f''(\xi)$, $\xi \in [a,b]$

số đoạn m tùy ý, bước tích phân $h = (b-a)/m$

Chương trình CT68 gồm các bước sau :

- Xác định hàm cần lấy tích phân bằng float ham (float x) trong đó hàm $y = e^{-x}\sin x$.

- Vào giá trị cận trên a, cận dưới b và số điểm tính m và xác định bước tích phân $h = (b-a)/m$.

- Dùng một chu trình for để tính tổng theo công thức Simpson.

```

/* CT68 Chuong trinh tinh tich phan xac dinh pp. Simpson */

#include<stdio.h>
#include<math.h>
float ham(float);

void main()
{
    int i,m;
    float a,b,x,h,x2,f2,x4,f4,tp;

    printf("\n Chuong trinh tinh tich phan pp Simpson");
    printf("\n Cho can duoi a = ") ;scanf("%f",&a);
    printf("\n Cho can tren b = ") ;scanf("%f",&b);
    printf("\n Cho so diem tinh m = ") ;scanf("%d",&m);
    h = (b-a)/m;
    x2 = a+h;
    x4 = a+h/2;
    f2 = ham(x2);
    f4 = ham(x4) ;
    for (i=1;i<m-2;i++)
    {
        x2 = x2+i*h;
        x4 = x4+i*h;
        f2 += ham(x2);
        f4 += ham(x4);
    }
    f2 = 2*f2;
    f4 = 4*(f4+ham(x4+h));
    tp= h*(f4+f2+ham(a)+ham(b))/6;
}

```

```

printf("\n Gia tri tich phan la TP= %8.3f",tp);
getch();
return;
} /* Ham can lay tich phan */

float ham(float x)
{
return exp(-x)*sin(x);
}

```

Kết quả của chương trình CT68 là :

Chương trình tính tích phân pp Simpson

Cho can duoi a = 1

Cho can tren b = 5

Cho so diem tinh m = 10

Gia tri tich phan la TP = 0.218

Bài số 69 Tính $\int_a^b f(x)dx$ bằng phương pháp hình thang.

Bài giải Cho hàm $y = f(x)$ liên tục trong khoảng $[a,b]$ tích phân xác định của hàm $y = f(x)$ trong khoảng $[a,b]$ được tính gần đúng bằng tổng các hình thang nguyên tố.

Giả thiết khoảng $[a,b]$ được chia thành m đoạn có bước tích phân $h=(b-a)/m$ thì $x_i = a + ih$, $i = 0, 1, \dots, m$.

Tích phân xác định được tính gần đúng bằng tổng các hình thang :

$$\int_a^b f(x)dx = h \left[\frac{f(a)+f(b)}{2} + \sum_{i=1}^{m-1} f(a+ih) \right] + R_m(f)$$

trong đó phần dư $R_m(f) = -\frac{(b-a)^3}{12m^2} f''(\xi)$, $\xi \in [a,b]$

Chương trình CT69 gồm các việc sau :

- Xác định hàm cần tinh tích phân bằng float ham (float x), trong bài này chọn hàm $y = e^x \sin x$.
- Dưa số liệu vào gồm cận dưới a, cận trên b, số điểm tính m.
- Dùng chu trình for để tính tổng giá trị tích phân theo phương pháp hình thang và ghi kết quả.

```
/* CT69 Chuong trinh tinh tích phan pp.hinh thang */
```

```
#include <stdio.h>
#include <math.h>
float ham(float);

void main()
{
    int i,m;
    float a,b,x,s,h,tp;
    printf("\n Chuong trinh tinh tích phan pp. hinh thang");
    printf("\n Cho can duoi a = ") ;scanf("%f",&a);
    printf("\n Cho can tren b = ") ;scanf("%f",&b);
    printf("\n Cho so diem tinh m = ") ;scanf("%d",&m);
    h = (b-a)/m;
    x = a;
    s = (ham(a)+ham(b))/2;
    for ( i = 1; i < m-1;i++)
    {
        x += i*h;
        s += 2*ham(x);
    }
    tp = s*h;
    printf("\n KET QUAT: %f",tp);
```

```

s += ham(x);
}
tp = s*h;
printf("\n Gia tri tich phan TP = %f",tp);
getch();
return;
} /* Ham can lay tich phan */

float ham(float x)
{
return exp(-x)*sin(x);
}

```

Ta được kết quả chương trình CT69 như sau :

Chương trình tính tích phân pp. hình thang

Cho can duoi a = 1

Cho can tren b = 5

Cho so diem tinh m = 10

Gia tri tich phan TP = 0.187901

Bài số 70 Tính tích phân kép $\sigma = \iint f(x,y) dx dy$ bằng phương pháp Gauss - Legendre ở n điểm.

Với $n \in (3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30)$. Các hoành độ u_1, \dots, u_n và các tâm w_1, \dots, w_n được tính toán trước.

Trong trường hợp $n = 3$ ta có :

$$\begin{array}{ll}
 u[1] = -0.774596 & w[1] = 0,555555 \\
 u[2]] = 0 & w[2] = 0,888888 \\
 u[3] = -u[1] & w[3] = w[1]
 \end{array}$$

$$\sigma' \approx \frac{h_x h_y}{4} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \sum_{l=1}^N \sum_{J=1}^N W_l W_J f\left(\frac{h_x}{2} u_l + x_l - v_2, \frac{h_y}{2} u_J + y_j - v_2\right)$$

Thành phần n_0 và các bước $h_x^{(0)} = b_1 - a_1$, $h_y^{(0)} = b_2 - a_2$ đặt trong các biến l kiểu nguyên và h_x , h_y kiểu thực. Thành phần q_0 kiểu thực.

Tiếp theo ở mỗi bước k ($1 \leq k \leq m$) thành phần l_k và các bước $h_x^{(k)} = (b_1 - a_1)/l_k$, $h_y^{(k)} = (b_2 - a_2)/l_k$ được đặt trong l và h_x , h_y , thành phần q_k đặt vào q_1 . Nếu $|q_k - q_{k-1}| < \delta |q_{k-1}|$ nội dung của Q_1 đưa vào biến tích phân tp rồi quá trình ngừng. Nếu không nội dung của Q_1 đưa vào Q_0 trước khi chuyển qua lần lặp tiếp theo.

Hàm cần lấy tích phân được định nghĩa trong float hàm (float x, float y) $f(x,y) = (x + y)\sin xy$.

```
/* CT70 Chuong trinh tich phan kep pp.Gauss-Legendre */
```

```
#include<stdio.h>
#include<math.h>
float ham(float,float);
```

```
void main()
```

```
{
```

```
int i,j0,k,l,m,it,n0;
```

```
float t,a1,b1,a2,b2,dta,hx,hy,q0,q1,s,x,x0,y,y0,tp;
```

```
float u[15],w[15];
```

```
printf("\nChuong trinh tich phan kep pp.Gauss-Legendre");
```

```
printf("\n VAO DU LIEU ");
```

```
printf("\n Cho can duoi a1 =") ; scanf("%f",&a1);
```

```
printf("\n Cho can tren b1 =") ; scanf("%f",&b1);
```

```
printf("\n Cho can duoi a2 =") ; scanf("%f",&a2);
```

```
printf("\n Cho can tren b2 =") ; scanf("%f",&b2);
```

```
printf("\n Do chinh xac dta=") ; scanf("%f",&dta);
```

```

printf("\n So lan lap max m =");scanf("%d",&m);
printf("\n So diem tinh toan no =3"); n0=3;
l=1;hx=b1-a1;hy=b2-a2;s=0;
x0=a1+hx/2;y0=a2+hy/2;
u[1]=-0.774596; w[1]=0.555555;
u[2]=0; w[2]=0.888888;
u[3]=-u[1]; w[3]=w[1];
for(i0=1;i0<=n0;i0++)
{
    x =hx*u[i0]/2+x0;
    for(j0=1;j0<=n0;j0++)
    {
        y=hy*u[j0]/2+y0;
        s=s+w[i0]*w[j0]*ham(x,y);
    }
}
q0=hx*hy/4;
it=-1; k=1;
while(it== -1 && k<=m)
{
    l=2*l;hx=hx/2;hy=hy/2;s=0;
    for(i=1;i<=l;i++)
    {
        x0=a1+(i-0.5)*hx;
        for(j=1 ;j<=l ;j++)
        {
            y0=a2+(j-0.5)*hy;
            for(i0=1 ;i0<=n0 ;i0++)

```

```

{
    x=hx*u[i0]/2+x0;
    for(j=1;j<=n0;j++)
    {
        y=hy*u[j]/2+y0;
        s=s+w[i0]*w[j]*ham(x,y);
    }
}
q1=hx*hy*s/4;
if( abs(q1-q0)<dta*abs(q0))
{
    tp=q1;it=1;
}
else
{
    q0=q1;k++ ;
}
printf(" \n Gia tri tich phan la TP =%f",tp);
getch();
return;
} /* Ham can lay tich phan */
float ham(float x,float y)
{
return (x+y)*sin(x)*sin(y);
}

```

Trong bài này nếu lấy miền $\mathbb{D} = [0, \pi]^2$ và $n_0 = 3$ ta được

Chuong trinh tich phan kep pp.Gauss - Legendre

VAO DU LIEU

Cho can duoi a1 = 0

Cho can tren b1 = 3.14159

Cho can duoi a2 = 0

Cho can tren b2 = 3.14159

Do chinh xac dta = 0.001

So lan lap max m = 10

So diem tinh toan no = 3

Gia tri tich phan la TP = 12.566351

Bài số 71 Tính tích phân kép $\sigma = \iint_{\mathbb{D}} f(x,y) dx dy$ bằng phương pháp Romberg.

Hàm $f(x)$ liên tục trong miền $\mathbb{D} = [a_1, b_1] \times [a_2, b_2]$. Vì $f(x)$ liên tục trong miền \mathbb{D} nên có thể viết :

$$\sigma = \int_{a_1}^{b_1} dx \int_{a_2}^{b_2} f(x,y) dy$$

Chia các cận $[a_1, b_1]$ và $[a_2, b_2]$ thành các đoạn nhỏ, nghĩa là chọn các bước tích phân h_x và h_y với các n_x và n_y là các số nguyên :

$$h_x = \frac{b_1 - a_1}{n_x}, \quad h_y = \frac{b_2 - a_2}{n_y}$$

giá trị của x và y tại các điểm lấy tích phân là

$$x_i = a_1 + i h_x, \quad i = 1, \dots, n_x$$

$$y_j = a_2 + j h_y, \quad j = 1, \dots, n_y$$

Miền \mathbb{D} được chia thành các miền P_{ij} với $P_{ij} = [x_{i-1}, x_i] \times [y_{j-1}, y_j]$

$$i = 1, \dots, n_x, \quad j = 1, \dots, n_y$$

Ta nhận được tích phân kép là một tổng như sau :

$$\sigma = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \iint_{P_{ij}} f(x,y) dx dy = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} \sigma_{ij}$$

Theo công thức điểm giữa ta có :

$$\sigma_{ij} \approx h_x h_y f(x_i - 1/2, y_j - 1/2)$$

$$\sigma \approx h_x h_y \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} f(x_i - 1/2, y_j - 1/2)$$

Chương trình tính tích phân kép trong miền D bằng phương pháp Romberg gồm các bước sau :

- Hàm cần lấy tích phân $f(x,y)$ được định nghĩa trong float ham (float x, float y).

- Đưa vào các giá trị cản tích phân a_1, b_1 và a_2, b_2 , độ chính xác dta và số lần lặp cực đại m. Ở mỗi giai đoạn k, $0 \leq k \leq m$, số n_k điểm, bước $h_x^{(k)} = (b_1 - a_1)/n_k$ của đoạn chia nhỏ $[a_1, b_1]$ và bước $h_y^{(k)} = (b_2 - a_2)/n_k$ của đoạn chia nhỏ $[a_2, b_2]$ được sắp xếp trong các biến nguyên l. Khi $k = 0$ thành phần t1 được đưa vào bảng t:

Nếu $k \geq 1$ bảng t chứa ở các vị trí tương ứng các thành phần $t_i^{(k-i)}$; $i = 1, \dots, k$. Thành phần $t_{k+1}^{(0)}$ được tính và đặt vào thành phần chỉ số $k + 1$ của t.

Nếu $|t_{k+1}^{(0)} - t_k^{(0)}| < \delta$ thành phần $t_{k+1}^{(0)}$ được đưa vào giá trị tích phân tp. Khi $it = -1$ quá trình ngừng, nếu không thành phần $t_i^{(k-i+1)}$, $i = k, \dots, 1$ được tính và đặt vào vị trí tương ứng của t. Ở mỗi giai đoạn i, số $2^{2(k-i+1)}$ ở trong C và nếu $|t_i^{(k-i+1)} - t_{i-1}^{(k-i+1)}| < \delta$ ($i \geq 2$) thành phần $t_i^{(k-i+1)}$ đưa vào tp, $it = 1$ và quá trình ngừng.

```
/* CT71 Chuong trinh tinh tich phan kep pp.Romberg */
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
float ham( float, float);
```

```

void main()
{
int i,it,j,k,l,m;
float a1,b1,a2,b2,dta,c,hx,hy,s,x,y,tp;
float t[15];
printf("\n Chuong trinh tinh tich phan kep pp. Romberg ");
printf("\n VAO DU LIEU ");
printf("\n Can duoi a1 = "); scanf("%f",&a1);
printf("\n Can tren b1 = "); scanf("%f",&b1);
printf("\n Can duoi a2 = "); scanf("%f",&a2);
printf("\n Can tren b2 = "); scanf("%f",&b2);
printf("\n Do chinh xac dta = "); scanf("%f",&dta);
printf("\n So lan lap max m = "); scanf("%d",&m);
l = 1; hx = b1-a1; hy = b2-a2;
t[1] = hx*hy*ham(a1+0.5*hx,a2+0.5*hy);
it = -1; k = 1;
while(it == -1 && k <= m)
{
    l = 2*l; hx = hx/2; hy = hy/2;
    s = 0;
    for (i=1;i<=l;i++)
    {
        x = a1+(i-0.5)*hx;
        for( j=1; j<l;j++)
        {
            y = a2+(j-0.5)*hy;
            s = s+ham(x,y);
        }
    }
}

```

```

    }

t[k+1] =hx*hy*s;

if (abs(t[k+1]-t[k])<dta)
{
    tp = t[k+1]; it = 1;
}

else
{
    c =1; i =k;
    while(it == -1 && i>=1)
    {
        c = 4*c ;
        t[i] = (c*t[i+1]-t[i])/(c-1) ;
        if (i>1)
            if( abs(t[i]-t[i-1])<dta)
                { tp =t[i] ;
                  it = 1 ;
                }
        i--;
    }
    k++;
}
printf("\n Gia tri tich phan la TP = %f ",tp);
getch();
return;
}

```

```
/* Ham can lay tich phan */
float ham( float x, float y)
{
    return (x+y)*sin(x)*sin(y);
}
```

Kết quả chương trình CT71 như sau :

Chuong trinh tinh tich phan kep pp.Romberg

VAO DU LIEU

Can duoi a1 = 0

Can tren b1 = 3.14159

Can duoi a2 = 0

Can tren b2 = 3.14159

Do chinh xac dta = 0.001

So lan lap max m = 10

Gia tri tich phan la TP = 12.520854

Chương 7

GIẢI PHƯƠNG TRÌNH PHI TUYẾN

Bài số 72 Tìm nghiệm của hàm phi tuyến trong khoảng $[a,b]$ bằng phương pháp chia đôi cung.

Bài giải Đầu tiên sơ bộ xác định khoảng nghiệm một cách hợp lý và tiến hành lặp lại quá trình tính toán sau đây :

- lấy khoảng giữa $m = (a+b)/2$
- tính toán $f(m)$
- nếu $f(m) = 0$, nghiệm là m
- nếu $f(a).f(m) < 0$ có một nghiệm trong khoảng $[a,m]$, thay thế khoảng $[a,b]$ bằng $[a,m]$ bằng cách đặt $b = m$
- nếu $f(a).f(m) > 0$ có một nghiệm trong khoảng $[b,m]$, thay thế $[a,b]$ bằng $[b,m]$ bằng cách đặt $a = m$.

Khi $| b-a |$ nhỏ hơn độ chính xác mong muốn quá trình tính toán ngừng và nghiệm gần đúng $f(m) \approx 0$.

Chương trình CT72 tìm nghiệm của hàm $\sin x$.

```
/* CT72 Chuong trinh tim cuc tri phuong phap chia doi cung */  
#include<stdio.h>  
#include<math.h>
```

```

int nghiem (double(*(double)(),double,double,double *,double);

void main()
{
double a,b, eps,z;
printf("\n Cho can duoi a = ");scanf("%f",&a);
printf("\n Cho can tren b = ");scanf("%f",&b);
printf("\n Cho do chinh xac eps = ");scanf("%f",&eps);
nghiem ( sin, a,b,&z,eps);
printf("\n Nghiem z = %f",z);
}

int nghiem(double (* f)(double),double a,
           double b,double *zero ,double eps)
{
double m, /* khoang dang xet */
      fm, /* gia tri ham fm */
      fa, /* giatri ham tai a */
      fb; /* gia tri ham tai b */
fa = (*f)(a);
fb = (*f)(b);
if(fa*fb >= 0 || a >= b) return(-1);
while (b-a > eps)
{
    m = (b+a)/2.0;
    fm = (*f)(m);
    if (fm == 0) break;
    if( fa*fm < 0 )
    {
        b = m;
    }
    else
        a = m;
}
*zero = m;
}

```

```

fb = fm;
}
else
{
    a = m;
    fa = fm;
}
}

* zero = m;
return(0);
}

```

Kết quả ta được :

Cho con dưới a = -1

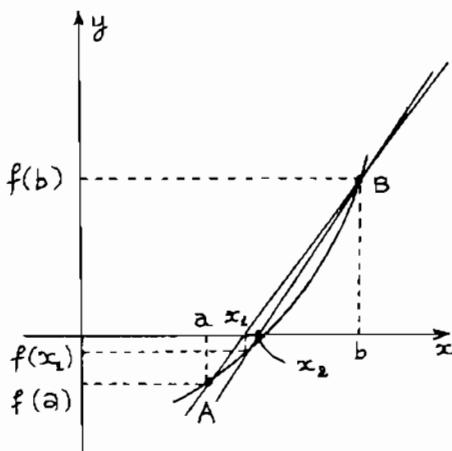
Cho con trên b = 3

Cho độ chính xác eps = 0.01

Nghiệm z = 0.000000

Bài số 73 Tìm nghiệm của phương trình phi tuyến $y = f(x)$ bằng phương pháp cắt tuyến.

Bài giải Xét đồ thị hàm số $y = f(x)$ hình 7-1. Giả thiết $f(a) < 0$ và $f(b) > 0$. Các điểm của đồ thị $A[a, f(a)]$ và $B[b, f(b)]$ được nối với nhau bằng một dây cung. Hoành độ x_1 của giao điểm dây cung AB và trục Ox là giá trị gần đúng của nghiệm tính theo công thức :



Hình 7-1

$$x_1 = a - \frac{(b-a)f(a)}{f(b) - f(a)}$$

trong đó x_1 thuộc khoảng $[a,b]$. Giả thiết $f(x_1) < 0$ khi đó được khoảng mới hẹp hơn là $[x_1,b]$. Nối các điểm $A_1[x_1,f(x_1)]$ và $B[b,f(b)]$ ta được giao điểm của dây cung với trục ox ở bước thứ hai theo công thức:

$$x_2 = x_1 - \frac{(b-x_1)f(x_1)}{f(b) - f(x_1)}$$

quá trình tính lặp tiếp tục ta được dãy số $a, x_1, x_2 \dots$ tiến dần tới nghiệm. Quá trình tính dừng khi sai số $|x_i - x_{i-1}| < \varepsilon$.

```
/* CT73 Chuong trinh giai pt phi tuyen pp. cat tuyen */
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define EPSI 0.005
float Ham(float);

void main()
{
    float a,b,fa,fb,x,dx;
    clrscr();
    printf("\n\t\t\tPHUONG PHAP CAT TUYEN GIAI HAM PHI
TUYEN");
    printf("\n\t\t\tHAM SO Y = EXP(-x)-x*x*x\n");
    printf("\nCho cac can x cua ham\n");
    printf("Cho a = ");scanf("%f",&a);
    printf("Cho b = ");scanf("%f",&b);
    fa = Ham(a);
    fb = Ham(b);
```

```

x = a;
dx = fa*(b-a)/(fa-fb);
while (abs(dx)>EPSI)
{
    x = a+dx;
    b = a;
    a = x;
    fb = fa;
    fa = Ham(x);
    dx = fa*(b-a)/(fa-fb);
}
printf("\n Nghiем x = %f",x);
printf("\n Giá trị ham = %f",Ham(x));
getch();
return;
}

float Ham(float x)
{
return(exp(x)+x*x*x);
}

```

Chương trình CT73 cho ta kết quả sau :

Cho các cần x của ham

Cho a = - 3

Cho b = 3

Nghiệm x = -0.815903

Giá trị ham = -0.100904

Bài số 74 Tìm nghiệm của hàm phi tuyến $y = f(x)$ bằng phương pháp lặp Newton.

Bài giải Để tìm nghiệm gần đúng của hàm $y = f(x)$ ta khai triển hàm thành chuỗi Taylor :

$$f(x) = f(x_n) + (x - x_n).f'(x_n) + \dots$$

Vì $f(\alpha) = 0$ suy ra :

$$\alpha = x_n - \frac{f(x_n)}{f'(x_n)} \text{ với } f'(x) \neq 0$$

Phương pháp lặp Newton tiến hành theo thuật toán :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, n=0,1,\dots$$

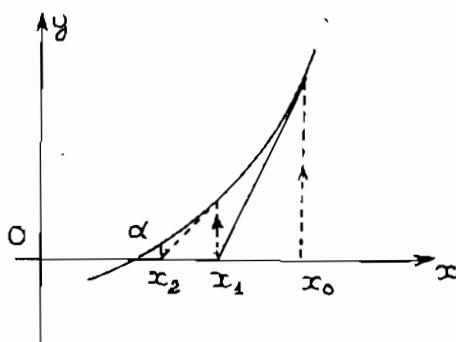
Về mặt hình học phương pháp lặp Newton (còn gọi là phương pháp tiếp tuyến) tiến hành tìm nghiệm gần đúng α với hoành độ x_α là điểm cắt của tiếp tuyến đường cong tại điểm x_{k-1} , đường cong được thay bằng đường tiếp tuyến (hình 7-2).

Phương pháp lặp sẽ hội tụ nếu $f(x)$ liên tục và có đạo hàm ở lân cận nghiệm α . Phép lặp sẽ dừng

khi sai số $|x_{n+1} - x_n| \leq \epsilon$.

Phương pháp lặp Newton hội tụ nhanh được coi là phương pháp cơ bản để tìm nghiệm của hàm phi tuyến.

Chương trình CT74 gồm việc xác định hàm $f(x)$ và đạo hàm cấp một $f'(x)$ và thực hiện một chu trình do while để tính lặp. Nếu bước lặp lớn hơn 100 sẽ báo bài toán không hội tụ.



Hình 7-2

```

/* CT74 Chuong trinh giai pt phi tuyen pp.Newton */

#include<stdio.h>
#include<math.h>
float Ham(float);
float Dham(float);

void main()
{
int i;
float x0,epsi,t;
float x[100];
printf("\n GIAI PT PHI TUYEN BANG PP. NEWTON ");
printf("\n Cho gia tri dau x0 =") ;scanf("%f",&x0);
printf("\n Cho sai so epsilon =") ;scanf("%f",&epsi);
i=1;
x[i]= x0;
do
{
    x[i+1] =x[i]-Ham(x[i])/Dham(x[i]);
    t = abs(x[i+1]-x[i]);
    x[i] = x[i+1]
    i++;
    if (i>100) printf("\n Bai toan khong hoi tu ");
}
while( abs(t) >epsi);
printf("\n Nghiem gan dung x =%f",x[i]);
printf("\n Gia tri ham = %f",Ham(x[i]));

```

```

getch();
return;
}

/* Ham can tim nghiem */
float Ham(float x)
{
    return(exp(-x)-x*x);
}

float Dham(float);
/* Dao ham */
float Dham(float x)
{
    return(-exp(-x)-2*x);
}

```

Ta đc kết quả :

Cho sai so epsilon = 0.001

Nghiem gan dung x = 0.733044

Gia tri ham = -0.056908

Chương 8

TỐI ƯU HÓA

Chương này trình bày một số thuật toán và chương trình tối ưu hóa các hàm một biến và nhiều biến. Đối với hàm một biến số thực đưa ra thuật toán và chương trình tìm cực trị theo phương pháp tiết diện và phương pháp gradient. Đối với hàm nhiều biến trình bày thuật toán và chương trình tìm cực trị hàm nhiều biến bằng phương pháp gradient.

Nếu hàm $f(x)$ xác định trong miền D thuộc không gian thực, hàm f có cực tiểu tổng thể tại điểm x_0 thuộc miền D nếu :

$$(\forall x \in D) \quad f(x) \geq f(x_0)$$

Hàm f có cực đại tổng thể tại điểm x_0 thuộc miền D nếu :

$$(\forall x \in D) \quad f(x) \leq f(x_0)$$

f có cực tiểu cục bộ ở điểm x_0 nếu tồn tại một điểm lân cận Y của x_0 sao cho :

$$(\forall x \in V \cap D) \quad f(x) \geq f(x_0)$$

và cực đại cục bộ tại x_0 sao cho :

$$(\forall x \in V \cap D) \quad f(x) \leq f(x_0)$$

Nếu hàm $f(x)$ có đạo hàm thì một số cực trị của nó có thể được xác định bằng cách tìm nghiệm phương trình $f'(x) = 0$. Tuy nhiên có những điểm tại đó đạo hàm bậc nhất bằng không không phải là điểm cực trị (điểm uốn) vì thế các kỹ thuật tìm kiếm cực trị của hàm số nói chung không dựa trên việc tìm nghiệm của phương trình $f'(x) = 0$.

Bài số 75 Tìm cực trị của hàm một biến $F(x)$ trong khoảng $[a, b]$ bằng phương pháp tiết diện vàng.

Bài giải Chuỗi q_k được coi là hội tụ về q , ta có :

$$\lim_{k \rightarrow +\infty} q_{k-1} = \lim_{k \rightarrow +\infty} q_k = q$$

và $q_k = 1 + \frac{1}{q_{k-1}}$, do đó ta có $q = 1 + \frac{1}{q}$, suy ra $q^2 - q - 1 = 0$
và có nghiệm dương là $\frac{\sqrt{5} + 1}{2} \approx 1,6180339887\dots$ gọi là số vàng.

$$\text{suy ra } r = \frac{1}{q} = q - 1 = \frac{\sqrt{5} - 1}{2} \approx 0,6180339887\dots$$

Phương pháp tiết diện vàng giữ tỷ số $\frac{l_{k+1}}{l_k}$ của hai khoảng liên tiếp
không đổi bằng cách lấy $\frac{l_{k+1}}{l_k} = r$, giữ nguyên quan hệ $l_k = l_{k+1} + l_{k+2}$,
 $k = 1, 2, \dots$

Ta có các biểu thức $l_{k+1} = rl_k$, $l_{k+2} = r^2l_1$, $l_k = ql_{k+1}$, $k = 1, 2, \dots$

Thuật toán tiến hành như sau :

Đặt

$$\left[\begin{array}{l} r = \frac{\sqrt{5} - 1}{2} \\ a_1 = a, b_1 = b, l_1 = b_1 - a_1 \\ l_2 = rl_1 \\ \alpha_1 = b_1 - l_2, \beta_1 = a_1 + l_2 \\ Y_{\alpha_1} = f(\alpha_1), Y_{\beta_1} = f(\beta_1) \end{array} \right]$$

$$\left[\begin{array}{ll} l_{k+2} = rl_{k+1} & \\ \text{nếu } Y_{\beta_k} < Y_{\alpha_k}, a_{k+1} = \alpha_k, b_{k+1} = b_k & \\ \quad \alpha_{k+1} = \beta_k, Y_{\alpha_{k+1}} = Y_{\beta_k} & \\ \quad \beta_{k+1} = a_{k+1} + l_{k+2}, Y_{\beta_{k+1}} = f(\beta_{k+1}) & | k=1,2,\dots \\ \text{nếu không, } a_{k+1} = a_k, b_{k+1} = \beta_k & \\ \quad \beta_{k+1} = \alpha_k, Y_{\beta_{k+1}} = Y_{\alpha_k} & \\ \quad a_{k+1} = b_{k+1} - l_{k+2}, Y_{\alpha_{k+1}} = f(\alpha_{k+1}) & \end{array} \right]$$

Ta cố định độ chính xác $\delta > 2\eta$ và ngừng quá trình khi $|l_{k+2}| \leq \delta$;

ta có $\begin{cases} a_{k+2} = \alpha_{k+1}, b_{k+2} = b_{k+1} & \text{nếu } Y_{\beta_{k+1}} < Y_{\alpha_{k+1}} \\ a_{k+2} = a_{k+1}, b_{k+2} = \beta_{k+1} & \text{nếu không} \end{cases}$

do đó ta có :

$$x_{\min} \approx (a_{k+2} + b_{k+2}) / 2 \text{ và } y_{\min} = f((a_{k+2} + b_{k+2}) / 2)$$

Chương trình gồm xác định hàm $f(x) = (4x^2 + 3x)e^{-2x}$ và đưa vào các cận :

a : cận dưới của khoảng tìm kiếm

b : cận trên của khoảng tìm kiếm, dta là sai số

```
/* CT75 Chuong trinh cuc tri theo tiet dien vang */
#include <stdio.h>
#include <math.h>
#include <conio.h>
float Ham(float );
void main ()
{
    float a,b,dta;
    float xmin,ymin;
    float l,r,to;
    float yalpha,ybeta,alpha,beta;
    clrscr();
    printf("\nTim cuc tri ham F(x) p p TIET DIEN VANG\n");
    printf("\nCho can duoi a = ");scanf("%f",&a);
    printf("\nCho can tren b = ");scanf("%f",&b);
    printf("\nCho sai so dta = ");scanf("%f",&dta);
    r = (sqrt(5)-1)/2;
```

```
l = b-a;
l = r*l;
alpha = b-l;
beta = a+l;
yalpha = Ham(alpha);
ybeta = Ham(beta);
do
{
    l = r*l;
    if (ybeta<yalpha)
    {
        a = alpha;
        alpha = beta;
        yalpha = ybeta;
        beta = a+l;
        ybeta = Ham(beta);
    }
    else
    {
        b = beta;
        beta = alpha;
        ybeta = yalpha;
        alpha = b-l;
        yalpha = Ham(alpha);
    }
    if (alpha>beta)
    {
        to = alpha;
```

```

alpha = beta;
rbeta = to;
tb = yalpha;
yalpha = ybeta;
ybeta = to;
}
}

while (l<=dta);
if (ybeta<yalpha) a = alpha;
else b = beta;
xmin = (a+b)/2;
ymin = Ham(xmin);
printf("\nKET QUA\t Xmin = %f\t Ymin = %f",xmin,ymin);
getch();
return;
}

/* Ham can tim cuc tri */
float Ham(float x)
{
return (4*x*x+3*x)*exp(-2*x) ;
}

```

Ta đc kết quả chương trình CT75 :

Tim cuc tri ham F(x) p p TIET DIEN VANG

Cho can duoi a = 0

Cho can tren b = 3

Cho sai so dta = 0.001

KET QUA Xmin = 2.427051 Ymin = 0.240466

Bài số 76 Tìm cực trị của hàm một biến $f(x)$ trong khoảng $[a, b]$ bằng phương pháp gradient.

Bài giải

Việc tìm cực trị của hàm một biến $f(x)$ dẫn tới tìm nghiệm của phương trình $f'(x) = 0$

Gọi h là bước tính $h = \frac{(b - a)}{n}$, n là số điểm tính toán ta có :

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

$$f''(x) \approx \frac{f(x+h) + f(x-h) - 2f(x)}{h^2}$$

Quá trình tính lặp theo phương pháp gradient cho theo công thức

$$x_{i+1} = x_i - h \frac{f(x+h) - f(x-h)}{2[f(x+h) + f(x-h) - 2f(x)]}$$

Chương trình tìm cực trị của hàm một biến theo phương pháp gradient gồm các bước sau :

- Đọc theo giá trị cận dưới a , bước h và sai số cho phép epsilon.
- Xây dựng hàm cần tìm cực trị, trong trường hợp bài toán có

$$f(x) = e^{-x} \sin x$$

- Tổ chức một chu trình do ... while để tính lặp. Chương trình dừng khi $|x_{i+1} - x_i| \leq \text{epsilon}$.

```
/* CT76 Chuong trinh cuc tri ham 1 bien */
```

```
#include <stdio.h>
```

```
#include <math.h>
```

```
float Ham(float );
```

```

void main ()
{
float a,h,eps,x,xe,ye,y1,y2,y3;
clrscr();
printf("\nTim cuc tri ham mot bien ");
printf("\nCho can duoi a = ");scanf("%f",&a);
printf("\nCho sai so eps ="); scanf("%f",&eps);
printf("\nCho buoc h = ");scanf("%f",&h)
do
{
    x = a;
    y1 = Ham(x-h);
    y2 = Ham(x);
    y3 = Ham(x+h);
    a = x-h*(y3-y1)/(2*(y3+y1-2*y2));
} while (abs(x-a) > eps);

xe = a;
ye = Ham(xe);
printf("\n Cuc tri tai xe = %f ,ye = %f",xe,ye);
getch();
return;
}

/* Ham can tim cuc tri */
float Ham(float x)
{
return exp(-x)*sin(x) ;
}

```

Kết quả chương trình CT76 như sau :

Tìm cực trị hàm một biến

Cho con duoi a = 0

Cho sai so eps = 0.001

Cho buoc h = 0.001

Cực trị tại xe = 0.499996, ye = 0.290785

Bài số 77 Tìm cực trị của hàm hai biến $f(x, y)$ bằng phương pháp gradient.

Bài giải Cho hàm hai biến $f(x, y)$, việc tìm cực trị của hàm $f(x, y)$ dẫn đến việc giải hệ phương trình:

$$f'_x(x,y) = 0 \text{ và } f'_y(x,y) = 0$$

Nếu $f(x, y)$ đã cho thì ta tính các đạo hàm riêng và bài toán tìm cực trị dẫn tới tìm nghiệm của hệ phương trình. Việc tính các đạo hàm cấp 2 thường dài, tuy nhiên ta có thể dùng một chu trình lặp theo phương pháp gradient như sau :

Trên mặt $f(x,y)$ ta tìm một cực tiểu theo hướng có độ dốc lớn nhất, nghĩa là theo hướng ∇f . Nếu $f(x,y)$ có cực tiểu tại x_m, y_m ta có quá trình lặp :

$$x_{i+1} = x_i - u_i f'_x(x_i, y_i)$$

$$y_{i+1} = y_i - u_i f'_y(x_i, y_i)$$

u_i có thể xác định ở mỗi bước tính.

Nếu chọn h là bước lặp các gradient theo phương pháp x và y được xác định như sau :

$$\text{grad}x = f(x+h, y) - f(x-h, y)$$

$$\text{grad}y = f(x, y+h) - f(x, y-h)$$

$$\text{grad} = (\text{grad}x^2 + \text{grad}y^2)^{1/2}$$

Quá trình tính lặp theo thuật toán sau :

$$x_{i+1} = x_i - l \cdot \text{grad}_x / \text{grad}$$

$$y_{i+1} = y_i - l \cdot \text{grad}_y / \text{grad}$$

trong đó l là khoảng cách cho trước.

Chương trình tìm cực trị của hàm hai biến theo phương pháp gradient gồm các bước sau :

- Định nghĩa hàm cần tìm cực trị, ở đây $f = (x+y)\sin x \cdot \sin y$
- Cho các cận dưới x_1, y_1 , bước h , khoảng cách l, l_{\max}
- Tạo chức chu trình tính lặp theo phương pháp gradient.

```
/* CT77 Chuong trinh tinh cuc tri pp.gradient */
```

```
#include<stdio.h>
#include<math.h>
float Ham(float,xl,float,yl);

void main()
{
int i,OK = 0;
float l,lmax,h,xl,yl,xm,ym,gradx,grady,grad;
printf("\n Tinh cuc tri ham 2 bien pp. gradient ");
printf("\n Cho can xl = "); scanf("%f",&xl);
printf("\n Cho can yl = "); scanf("%f",&yl);
printf("\n Cho buoc h = "); scanf("%f",&h);
printf("\n Cho khoang cach l= ");scanf("%f",&l);
printf("\n Cho lmax = "); scanf("%f",&lmax);
xm = xl,ym = yl;
```

```

for(i=0;!OK;i++)
{
    if(i> 1000)
        { OK = 1;
        printf("\n Tinh toan qua dai !\n ");
        }

    gradx = Ham(xm+h,ym)-Ham(xm-h,ym);
    grady = Ham(xm,ym+h)-Ham(xm,ym-h);
    grad = sqrt(gradx*gradx+grady*grady);
    xl = xm, yl = ym;
    l *= 2;
    if( grad!= 0)
    {
        do
        {
            l /= 2;
            xm = xl-l*gradx/grad;
            ym = yl-l*grady/grad;
            if(l<lmax) OK = 1;
        }
        while(Ham(xm,ym) < Ham(xl,yl));
    }
    else OK = 1;
}

printf("\n Cuc tri tai xm =%f ym =%f",xm,ym);
getch();
return;
}

```

```
/* Ham can tim cuc tri */
float Ham(float x, float y)
{
    return(x+y)*sin(x)*sin(y);
}
```

Kết quả chương trình CT77 như sau :

Tính cực trị hàm 2 biến pp. gradient

Cho can xl = -2

Cho can yl = 2

Cho buoc h = 0.001

Cho khoang cach l = 10

Cho lmax = 15

Cực trị tại xm = -0.232233 ym = 3.767767

Chương 9

PHƯƠNG TRÌNH VI PHÂN

Trong chương này trình bày một số thuật toán và chương trình giải gần đúng phương trình vi phân cấp một và hệ n phương trình vi phân cấp một bài toán sơ kiện (Bài toán Cauchy).

Cho phương trình vi phân (PTVP) cấp một dạng :

$$y' = f(x,y) \quad (1)$$

Ta gọi nghiệm của phương trình (1) trong khoảng I là hàm y có đạo hàm trong khoảng I sao cho $\forall x \in I$ có $y'(x) = f(x, y(x))$.

Việc giải phương trình (1) là tìm các nghiệm theo sơ kiện sao cho phương trình vi phân có một giá trị cố định tại một điểm cho trước.

$$\begin{cases} y' = f(x,y) \\ y(a) = y_a \end{cases}, \quad \forall x \in [a,b] \quad (2)$$

Nếu f liên tục trong khoảng $[a, b]$, thỏa mãn điều kiện Lipschitz :

$$\forall x \in [a, b] \quad \forall u, v \in \mathbb{R} \text{ thì } |f(x, u) - f(x, v)| \leq L |u - v|, \quad L > 0$$

phương trình vi phân (2) có một nghiệm duy nhất là y.

Nghiệm bằng số của PTVP (2) là dãy số y_1, y_2, \dots, y_N là các giá trị gần đúng của các giá trị chính xác $y(x_1), y(x_2), \dots, y(x_N)$ của nghiệm chính xác y ở N điểm $x_1 < x_2 < \dots < x_N$ của khoảng $[a, b]$ với bước $h_i = x_{i+1} - x_i$, $i = 0, \dots, N-1$ với $x_0 = a$ có thể là hằng số hoặc không.

Các phương pháp giải gần đúng có bước không đổi được đặc trưng bằng việc để tính toán một giá trị mới y_{i+1} chỉ cần thông tin tại một điểm (x_i, y_i) .

Các phương pháp có bước thay đổi được đặc trưng bằng việc tính toán ở giá trị mới y_{i+1} cần thông tin ở nhiều bước trước và một quá trình ngoại suy.

Bài số 78 Giải phương trình vi phân cấp một $y' = f(x,y)$ thỏa mãn sơ kiện $y(x_0) = y_0$ bằng phương pháp Euler.

Bài giải

Phương pháp Euler là phương pháp gần đúng có bước h không đổi theo công thức lặp :

$$y_{i+1} = y_i + hf(x_i, y_i), i = 0, \dots, N-1$$

Sai số của phương pháp cỡ h^2 .

Để tổ chức chương trình giải PTVP cấp một theo phương pháp Euler CT78 gồm các bước sau :

- Xác định hàm $y' = f(x,y)$
- Cho khoảng tích phân $[a,b]$ và sơ kiện x_0, y_0
- Tổ chức tính lặp theo công thức Euler và ghi kết quả các giá trị $y_i(x_i)$.

```
/* CT78 Chuong trinh giai PTVP pp. EULER */
```

```
#include<stdio.h>
#include<math.h>
float ham(float);
void main()
{
    int i,m;
```

```

float a,b,h,x0,y0,c1,c2;
float x[100],y[100];
printf("\n Giai PTVP pp.Euler ");
printf("\n Vao du lieu");
printf("\n Cho gia tri dau a =");scanf("%f",&a);
printf("\n Cho gia tri cuoi b=");scanf("%f",&b);
printf("\n Cho so buoc m =");scanf("%d",&m);
printf("\n Cho so kien x0 =");scanf("%f",&x0);
printf("\n Cho so kien y0 =");scanf("%f",&y0);
h=(b-a)/m;
x[0] = x0;
y[0] = y0;
printf("\n Bang ket qua ");
for( i=0; i<=m;i++)
{
    x[i+1] = x[i]+h;
    y[i+1] = y[i]+h*ham(x[i]);
    printf("\n %-10.3f %-10.3f",x[i],y[i]);
}
getch();
return;
}


```

```

/* Ham can giao */
float ham(float x)
{
    return exp(-x)*sin(x);
}

```

Ta được bảng kết quả sau đây :

Giai PTVP pp.Euler

Vào du lieu

Cho giá trị đầu a = 0

Cho giá trị cuối b = 2

Cho số bước m = 10

Cho số kien x0 = 0

Cho số kien y0 = 0

Bảng kết quả

0.000 0.000

0.200 0.000

0.400 0.033

0.600 0.085

0.800 0.147

1.000 0.211

1.200 0.273

1.400 0.329

1.600 0.378

1.800 0.418

2.000 0.450

Bài số 79 Giải phương trình vi phân cấp một $y' = f(x,y)$ thỏa mãn
số kiện $y'(x_0) = y_0$ bằng phương pháp Euler cải tiến.

Bài giải

Phương pháp Euler cải tiến là phương pháp gần đúng có bước h
không đổi theo công thức lặp sau đây :

$$\left| \begin{array}{l} C_1 = hf(x_i, y_i) \\ C_2 = hf(x_i + h, y_i + C_1) \\ y_{i+1} = y_i + \frac{1}{2} (C_1 + C_2) \end{array} \right| \quad i = 0, \dots, N-1$$

sai số của phương pháp cỡ h^3 :

Chương trình CT79 gồm các bước sau đây :

- Xác định hàm $y' = f(x, y)$
- Cho khoảng tích phân $[a, b]$ và sơ kiện $y'(x_0) = y_0$
- Tổ chức tính lặp theo phương pháp Euler cài tiến.

/ CT79 Chương trình giải PTVP pp. Euler cài tiến */*

```
#include<stdio.h>
#include<math.h>
float ham(float);

void main()
{
    int i,m;
    float a,b,x0,y0,h,c1,c2;
    float x[100],y[100];
    printf("\n Vao du lieu");
    printf("\n Gia tri dau a = ");scanf("%f",&a);
    printf("\n Gia tri cuoi b = ");scanf("%f",&b) ;
    printf("\n So buoc m = "); scanf("%d",&m);
    printf("\n So kien xo ="); scanf("%f",&x0);
    printf("\n So kien y0 = "); scanf("%f",&y0);
    h = (b-a)/m;
```

```

x[0] = x0;
y[0] = y0;
printf("\n Bang ket qua");
for( i=0; i<=m;i++)
{
    x[i+1] = x[i]+h;
    c1 = h*ham(x[i]);
    c2 = h* ham(x[i]+h);
    y[i+1] =y[i]+(c1+c2)/2;
    printf("\n %-10.3f %-10.3f",x[i],y[i]);
}
getch();
return;
}

float ham(float x)
{
return exp(-x)*sin(x);
}

```

Kết quả ta được bảng bảng giá trị y_i (x_i).

Vào dữ liệu

Giá trị đầu a = 0

Giá trị cuối b = 2

Số bước m = 10

Số kiêm x0 = 0

Số kiêm y0 = 0

Bang ket qua

0.000	0.000
0.200	0.016
0.400	0.059
0.600	0.116
0.800	0.179
1.000	0.242
1.200	0.301
1.400	0.354
1.600	0.398
1.800	0.434
2.000	0.463

Bài số 80 Giải phương trình vi phân cấp một $y' = f(x,y)$ thỏa mãn sơ kiện $y'(x_0) = y_0$ bằng phương pháp Runge - Kutta.

Bài giải Nghiệm gần đúng của PTVP cấp một trong khoảng $[a,b]$, bước $h = x_{i+1} - x_i$ thỏa mãn sơ kiện $y'(x_0) = y_0$ được tìm theo công thức Runge - Kutta cấp 4 :

$$\left[\begin{array}{l} C_1 = hf(x_i, y_i) \\ C_2 = hf\left(x_i + \frac{1}{2} h, y_i + \frac{1}{2} C_1 \right) \\ C_3 = hf\left(x_i + \frac{1}{2} h, y_i + \frac{1}{2} C_2 \right) \\ C_4 = hf\left(x_i + h, y_i + C_2 \right) \\ y_{i+1} = y_i + \frac{1}{6} (C_1 + 2C_2 + 2C_3 + C_4) \end{array} \right] \quad i = 0, 1, \dots, N-1$$

Sai số của phương pháp cỡ h^5 .

Chương trình CT80 gồm các bước sau :

- Xác định hàm $y' = f(x,y)$
- Cho khoảng tích phân $[a,b]$ và sơ kiện x_0, y_0
- Tính lặp theo công thức Runge - Kutta cấp 4.

Ta nhận thấy sai số của phương pháp giảm đi khi số điểm tính N tăng lên, nghĩa là bước h nhỏ đi. Tuy nhiên các phương pháp có bước không đổi như Euler, Euler cải tiến và Runge - Kutta có nhược điểm là sai số gấp phải ngoài sai số phương pháp còn phải kể tới sai số làm tròn của máy tính, khi số điểm tính N tăng sai số phương pháp giảm đi nhưng sai số làm tròn tăng lên. Do đó đối với mỗi phương pháp tính có một giá trị N tối ưu không được vượt quá. Giá trị N phụ thuộc vào hàm f và vào máy tính.

/* CT80 Chương trình giải PTVP pp.Runge-Kutta */

```
#include<stdio.h>
#include<math.h>
float ham(float);

void main()
{
    int i,m;
    float a,b,x0,y0,h,k1,k2,k3,k4;
    float x[100],y[100];
    printf("\n Phuong phap RK 4");
    printf("\n Vao du lieu ");
    printf("\n Gia tri dau a = ");scanf("%f",&a);
    printf("\n Gia tri cuoi b= ");scanf("%f",&b);
    printf("\n So buoc tinh m= ");scanf("%d",&m);
```

```

printf("\n So kien x0 = ");scanf("%f",&x0);
printf("\n So kien y0 = ");scanf("%f",&y0);
h = (b-a)/m;
x[0] =x0;
y[0] =y0;
printf("\n Bang ket qua ");
for (i=0;i<=m;i++)
{
    x[i+1] = x[i]+h;
    k1 = h*ham(x[i]);
    k2 = h*ham(x[i]+h/2);
    k3 = h*ham(x[i]+h/2);
    k4 = h*ham(x[i]+h);
    y[i+1] = y[i]+(k1+2*k2+2*k3+k4)/6;
    printf("\n %-10.3f %-10.3f",x[i],y[i]);
}
getch();
return;
}

float ham(float x)
{
return exp(-x)*sin(x) ;
}

```

Ta được kết quả chương trình CT80 :

Phương pháp RK = 4

Vào điểm

Gia trị đầu a = 0

Gia tri cuoi b = 2

So buoc tinh $m = 10$

So kien $x_0 = 0$

So kien $y_0 = 0$

Bang ket qua

0.000	0.000
0.200	0.017
0.400	0.061
0.600	0.119
0.800	0.182
1.000	0.246
1.200	0.305
1.400	0.358
1.600	0.402
1.800	0.438
2.000	0.467

Bài số 81 Giải hệ phương trình vi phân cấp n thỏa mãn sơ kiện $y(0)=y_0$

Bài giải Phương trình vi phân (PTVP) cấp n có thể đưa về hệ n PTVP cấp một dạng :

$$\left\{ \begin{array}{l} y_1 = g_1(x, y_1, \dots, y_n) \\ y_2 = g_2(x, y_1, \dots, y_n) \\ \dots \\ y_n = g_n(x, y_1, \dots, y_n) \end{array} \right. \quad (1)$$

trong đó : g_1, \dots, g_n là các hàm của x, y_1, y_2, \dots, y_n

y_1, \dots, y_n là các hàm biến số thực của x

y'_1, \dots, y'_n là các đạo hàm cấp một của các hàm y_1, \dots, y_n

Ta gọi nghiệm của hệ PTVP cấp một trong khoảng I của không gian thực R là các hàm $y = (y_1, \dots, y_n)$ có đạo hàm trong khoảng I sao cho với mọi x thuộc I ta có :

$$y'(x) = g(x, y(x))$$

Giải hệ PTVP (1) là tìm toàn bộ các nghiệm này. Giải hệ PTVP (1) theo các điều kiện đầu :

$$\begin{cases} y' = g(x, y) \\ y(a) = Y_a \end{cases} \quad x \in [a, b] \quad (2)$$

nếu g liên tục trong khoảng $[a, b]$ và thỏa mãn điều kiện Lipschitz thì (2) có nghiệm duy nhất. Nghiệm số của hệ (2) là một dãy các vectơ Y_1, \dots, Y_N làm gần đúng các giá trị $y(x_1), \dots, y(x_N)$ của nghiệm chính xác y ở N điểm $x_1 < x_2 < \dots < x_N$ của khoảng $[a, b]$ với bước $h_i = x_{i+1} - x_i$, $i = 0, \dots, N-1$ và $x_0 = a$.

Ta dùng thuật toán tính lặp theo Runge - Kutta cấp 4 như sau :

$C_1 = hg(x_i, Y_i)$ $C_2 = hg(x_i + \frac{1}{2} h, Y_i + \frac{1}{2} C_1)$ $C_3 = hg(x_i + \frac{1}{2} h, Y_i + \frac{1}{2} C_2)$ $C_4 = hg(x_i + h, Y_i + C_3)$ $Y_{i+1} = Y_i + \frac{1}{6} (C_1 + 2C_2 + 2C_3 + C_4)$	$i = 0, 1, \dots, N-1$
---	------------------------

Trong chương trình CT81 hệ PTVP cấp n được xác định bằng hàm G có dạng :

`void G (int n, double x, double * y, double * w)`

và được minh họa bằng hệ 2 PTVP cấp một :

$$y'_1 = 1 - 2y_1/x$$

$$y'_2 = y_1 - y_2 - 1 + 2y_1/x$$

với $x \in [1,2]$ thỏa mãn sơ kiện :

$$y_1(x_0) = y_1(1) = 1/3 \text{ và}$$

$$y_2(x_0) = y_2(1) = -1/3$$

Chương trình CT81 gồm các bước :

- Xác định hàm G là hệ n PTVP cấp một cần giải.
- Cho sơ kiện bài toán.
- Cho bước tích phân h và khoảng nghiệm [a,b]
- Dùng chu trình for để tìm nghiệm gần đúng của hệ PTVP (2) theo phương pháp Runge - Kutta cấp 4 và in các kết quả :

$$y_1(x_1), \dots, y_1(x_n)$$

$$y_2(x_1), \dots, y_2(x_n)$$

.....

$$y_n(x_1), \dots, y_n(x_n)$$

```
/* CT81 Chuong trinh giai he PTVP cap n pp Runge-Kutta 4 */
```

```
#include<stdio.h>
```

```
#include<math.h>
```

```
#define N 21
```

```
typedef double vec[N];
```

```
vec c1,c2,c3,c4,y,v,w;
```

```
void G(int,float,double*,double*);
```

```
void main()
```

```
{
```

```
int i,k,n,l;
```

```
float x,x0,h,t;
```

```
printf("\n Cho so phuong trinh n = ");scanf("%d",&n);
```

```

printf("\n Cho buoc h = "); scanf("%f",&h);
printf("\n Cho so kien x0 = ");scanf("%f",&x0);
/* Phuong phap Runge-Kutta cap 4 */
for(l=1;l<=n;l++)
{
    printf("\n Cho so kien y[%d]= ",l);
    scanf("%f",&t);y[l]= t;
}
x = x0;
for( i=1;i<=100;i++)
{
    {G(n,x,y,w);
    for(k=1;k<=n;k++)
    {
        c1[k] = h*w[k];
        v[k] = y[k]+c1[k]/2;
    }
    G(n,x+h/2,v,w);
    for(k=1;k<=n;k++)
    {
        c2[k] = h*w[k];
        v[k] = y[k]+c2[k]/2;
    }
    G(n,x+h/2,v,w);
    for(k=1;k<=n;k++)
    {
        c3[k] = h*w[k];
        v[k] = y[k]+c3[k];
    }
    G(n,x+h,v,w);
}

```

```

for(k=1;k<=n;k++)
{
    c4[k] = h*w[k];
    y[k]=y[k]+(c1[k]+2*c2[k]+2*c3[k]+c4[k])/6;
    printf("\n x = %f y[%d] = %f\n",x,k,y[k]);
}
x += h;
}

getch();
return;
}

void G( int n,float x,double *y,double *w) /* PTVP dang chuan */
{
w[1] = 1-2*y[1]/x;
w[2] = y[1]+y[2]-1+2*y[1]/x;
}

```

Ta được kết quả chương trình như sau :

Cho số phuong trình n = 2

Cho buoc h = 0.01

Cho so kien x0 = 1

Cho so kien y1 = 0.333

Cho so kien y2 = -0.333

<i>x = 1.000000</i>	<i>y1 = 0.333333</i>	<i>y2 = -0.333333</i>
---------------------	----------------------	-----------------------

<i>x = 1.100000</i>	<i>y1 = 0.366666</i>	<i>y2 = -0.366666</i>
---------------------	----------------------	-----------------------

<i>x = 1.300000</i>	<i>y1 = 0.433333</i>	<i>y2 = -0.433333</i>
---------------------	----------------------	-----------------------

<i>x = 1.700000</i>	<i>y1 = 0.566666</i>	<i>y2 = -0.566666</i>
---------------------	----------------------	-----------------------

<i>x = 2.000000</i>	<i>y1 = 0.666666</i>	<i>y2 = -0.666666</i>
---------------------	----------------------	-----------------------

Chương 10

XÁC SUẤT THỐNG KÊ

Bài số 82 Tìm phân bố nhị thức biết số phép thử $n = 10$, xác suất xuất hiện biến cố ngẫu nhiên trong mỗi phép thử $p = 0,45$, $x = 4$.

Bài giải Biến ngẫu nhiên x có thể nhận các giá trị nguyên $0, 1, 2, \dots, n$. Xác suất để trong n phép thử biến cố xuất hiện x lần được tính theo công thức Bernouille :

$$P_{x,n} = C_n^x p^x q^{n-x}, \text{ trong đó } q = 1-p$$

Tính tổ hợp chập x trong n phần tử được tính theo công thức :

$$C_n^x = \frac{n!}{x!(n-x)!}$$

do đó ta được :

$$P_{x,n} = \frac{n!}{x!(n-x)!} p^x (1-q)^{n-x}$$

Chương trình phân bố nhị thức gồm các bước sau :

- Đọc số lần thử $n = 10$, xác suất $p = 0,45$, biến cố xuất hiện $x=4$ sau đó gọi hàm tính giai thừa.

- Đặt $s_1 = n!$, $s_2 = x!$, $s_3 = (n-x)!$
- Dùng hàm pow để tính các hàm mũ : $s_4 = \text{pow}(p,x)$
 $s_5 = \text{pow}(1-p, n-x)$

Kết quả ta được phân bố nhị thức :

$$B = \frac{s_1}{s_2 \cdot s_3} \cdot s_4 \cdot s_5$$

```
/* CT82 Chuong trinh tinh phan bo nhi thuc */
#include<stdio.h>
#include<math.h>
float GIAITHUA(int);

void main()
{
int n,x;
float p,s1,s2,s3,s4,s5,nt;
printf("\n So phep thu n = "); scanf("%d",&n);
printf("\n So lan xuat hien bien co x = ");scanf("%d",&x);
printf("\n Xac suat xuat hien x p = ");scanf("%f",&p);
s1 = GIAITHUA(n);
s2 = GIAITHUA(x);
s3 = GIAITHUA(n-x);
s4 = pow(p,x);
s5 = pow(1-p,n-x);
nt = s1*s4*s5/(s2*s3);
printf("\n Phan bo nhi thuc : %f ",nt);
getch();
return;
}
```

```

/* Tinh giai thua */
float GLAITHUA(int n)
{
    if(n==1 || n==0)
        return(1);
    else
        return(n*GLAITHUA(n-1));
}

```

Kết quả chương trình CT82 như sau :

So phep thu n = 3

So lan xuat hien bien co x = 4

Xac suat xuat hien x p = 0.45

Phan bo nhi thuc : 0.238367

Bài số 83 Dùng phân bố Poisson để tính trong 1000 trang sách có 100 lỗi in sai. Tính xác suất để khi lấy ngẫu nhiên 1 trang có không quá 4 lỗi.

Bài giải Phân bố Poisson cho bởi biểu thức :

$$P(x,y) = \frac{a^m}{m!} e^{-a}$$

trong bài này $a = \frac{100}{1000} = 0,1$, $m = 4$

Chương trình gồm các bước sau :

- Đọc từ bàn phím số lỗi $m = 4$, $a = 0,1$
- Xây dựng hàm GLAITHUA (m)

- Dùng hàm mű pow để tính a^m
- Tính phân bố Poisson theo công thức $\frac{a^m}{m!} e^{-a}$

```

/* CT83 Chuong trinh tinh phan bo Poisson */
#include<stdio.h>
#include<math.h>
float GIAITHUA(int);

void main()
{
int m;
float a,P;
printf("\n Cho m = ");scanf("%d",&m);
printf("\n Cho a = ");scanf("%f",&a);
P =pow(a,m)*exp(-a)/GIAITHUA(m);
printf("\n Phan bo Poisson P = %f",P);
getch();
return;
}

/* Ham giai thua */
float GIAITHUA(int n)
{
if(n==1||n==0)
    return(1);
else
    return(n*GIAITHUA(n-1));
}

```

Chương trình CT83 cho kết quả như sau :

Cho $m = 4$

Cho $a = 0.1$

Phân bố Poisson $P = 0.000004$

Bài số 84 Biến ngẫu nhiên x_i có xác suất p_i cho theo bảng phân bố x_i , p_i . Tìm kỳ vọng toán học, phương sai, độ lệch chuẩn phương của các biến với số a cho trước bất kỳ.

Bài giải Kỳ vọng toán học của các biến ngẫu nhiên được xác định theo công thức :

$$M(x) = \sum_{i=1}^n x_i p_i$$

trong đó x_i là biến thứ i , p_i là xác suất xuất hiện biến x_i .

Phương sai : $D(x) = M[x - M(x)]^2 = M[(x-a)^2] - (M-a)^2$

Độ lệch chuẩn phương : $\sigma_x = \sqrt{D(x)}$

Chương trình gồm các bước sau :

- Đọc bảng $x[i]$, $p[i]$, số a , n , từ bàn phím.

- Tính kỳ vọng toán học $M(x)$ theo thuật toán tính một tổng (xem CT23).

- Tính $D_1 = \sum_{i=1}^n p_i (x_i - a)^2$ và tính phương sai $D(x)$ theo thuật toán tính một tổng!

- Cuối cùng tính độ lệch chuẩn phương và ghi kết quả.

/ CT84 Chương trình tính kỳ vọng và phương sai */*

```
#include <stdio.h>
```

```
#include <math.h>
```

```

void main()
{
int i,n;
float d,d1,m,a,sigma;
float x[100],p[100];
printf("\n Cho so lan thu n = ");scanf("%d",&n);
printf("\n Cho so a = ");scanf("%f",&a);
printf(" Vao bang x[i] ");
for(i=1;i<=n;i++)
{
    printf("\nx[%d] = ",i);scanf("%f",&x[i]);
    printf("\n p[%d] = ",i); scanf("%f",&p[i]);
}
m = 0;
d1 = 0;
for (i=1;i<=n;i++)
{
    m += x[i]*p[i];
    d1 += p[i]*(x[i]-a)*(x[i]-a);
    d = abs(d1-(m-a)*(m-a));
    sigma = sqrt(d);
}
printf("\n Ky vong toan hoc m = %f",m);
printf("\n Phuong sai d = %f",d);
printf("\n Do lech quan phuong s =%f",sigma);
getch();
return;

```

Kết quả chương trình nếu cho bảng giá tr

$$\begin{array}{ll} x[1] = 1 & p[1] = 0.2 \\ x[2] = 2 & p[2] = 0.3 \\ x[3] = -1 & p[3] = 0.3 \\ x[4] = 2.5 & p[4] = 0.4 \\ x[5] = 1.5 & p[5] = 0.1 \\ a = 2 \end{array}$$

ta tìm được :

$$Ky vong toan hoc \quad m = 1.650000$$

$$Phuong sai \quad d = 2.000000$$

$$Do lech quan phuong \quad s = 1.414214$$

Bài số 85 Xác định hệ số tương quan giữa hai biến ngẫu nhiên x và y. Kiểm tra xem mối liên hệ giữa x và y có đủ tin cậy không.

Bài giải Hai biến ngẫu nhiên x và y dạng vectơ n phần tử $x(i)$ và $y(i)$, $0 \leq i \leq n$ có tương quan với nhau. Để xác định hệ số tương quan giữa hai biến ta áp dụng công thức :

$$R_{xy} = \frac{C_{xy}}{\sigma_x \cdot \sigma_y}$$

trong đó : $C_{xy} = \frac{1}{n} \sum_{i=1}^n x_i y_i - \bar{x} \bar{y}$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$
$$\sigma_x^2 = \frac{1}{n} \sum_{i=1}^n (x_i^2 - \bar{x}^2), \quad \sigma_y^2 = \frac{1}{n} \sum_{i=1}^n (y_i^2 - \bar{y}^2)$$

và kiểm tra điều kiện $| R_{xy} | \cdot \sqrt{n-1} \geq 3$ thì liên hệ giữa các biến $x(i)$ và $y(i)$ là tin cậy.

Chương trình gồm các công việc sau :

- Đọc các vectơ $x(i)$ và $y(i)$ từ bàn phím.
- Tính trung bình cộng \bar{x} và \bar{y} theo thuật toán tính một tổng.
- Tính các độ lệch chuẩn phương σ_x và σ_y .
- Tính hệ số tương quan R_{xy} và kiểm tra điều kiện liên hệ tin cậy

$$| R_{xy} | \cdot \sqrt{n-1} \geq 3$$

```
/* CT85 Chương trình tính hệ số tương quan */
```

```
#include<stdio.h>
```

```
#include<math.h>
```

```
void main()
```

```
{
```

```
int i,n;
```

```
float t,r,c,cxy,cxy1,sigmax,sigmay,rxy,
```

```
    sigmax1,sigmay1,x1,y1,xtb,ytb;
```

```
float x[100],y[100];
```

```
printf("\n He so tuong quan giua hai bang du lieu ");
```

```
printf("\n Vao du lieu ");
```

```
printf("\n Cho so bien n = "); scanf("%d",&n);
```

```
printf("\n Vao bang x[i],y[i]");
```

```
for(i=1;i<=n;i++)
```

```
{
```

```
printf("\n x[%d ] = ",i);
```

```
scanf("%f",&x[i]);
```

```
printf("\n y[%d ] = ",i);
```

```
scanf("%f",&y[i]);
```

```
}
```

```

x1=0 ;y1=0;sigmax=0;sigmay=0;cxy=0;
for(i=1;i<=n;i++)
{
    x1=x1+x[i];
    y1 = y1+y[i];
}
xtb = x1/n;ytb = y1/n;
printf("xtb = %f ytb = %f",xtb,ytb);
for(i=1;i<=n;i++)
{cxy = cxy+x[i]*y[i];}
cxy1 = cxy/n-xtb*ytb;
printf("\n cxy1 = %f ", cxy1);
for(i=1;i<=n;i++)
{ sigmax = sigmax+x[i]*x[i]-xtb*xtb;}
sigmax1 = sqrt(abs(sigmax/n));
printf("\n sigmax1 = %f",sigmax1);
for(i=1;i<=n;i++)
{ sigmay = sigmay+y[i]*y[i]-ytb*ytb;}
sigmay1 = sqrt(abs(sigmay/n));
printf("\n sigmay1 = %f",sigmay1);
rxy =cxy1/(sigmax1+sigmay1);
t = sqrt(n-1);
if(abs(rxy*t)>=3)
printf("\n Lien he tin cay rxy = %f",rxy);
else printf("\n Lien he khong tin cay rxy = %f",rxy);
getch();
return;
}

```

Kết quả chương trình nếu cho các bảng giá trị : n = 5

$x[1] = 1$ $y[1] = 2$

$x[2] = -1$ $y[2] = 3$

$x[3] = 0$ $y[3] = -1$

$x[4] = 3$ $y[4] = 1$

$x[5] = 2$ $y[5] = 2$

$xtb = 1.000000$ $ytb = 1.400000$

$cxy1 = -0.200000$

$sigmax1 = 1.414214$

$sigmay1 = 1.000000$

Lien hệ không tin cay rxy = -0.082843

Chương 11

MỘT SỐ CHƯƠNG TRÌNH XỬ LÝ VĂN BẢN, QUẢN LÝ VÀ ỨNG DỤNG

Chương này trình bày một số chương trình xử lý văn bản, quản lý và một số bài toán ứng dụng của ngôn ngữ C. Phần giải thích chương trình đã được đưa vào văn bản chương trình nên lời giải thích dài trong chương này là không cần thiết.

Bài số 86 Lập trình in tờ lịch thế kỷ.

Bài giải Để có một tờ lịch cần các dữ liệu sau :

Thế kỷ, năm, tháng, số ngày trong tuần

Lịch chia làm hai phần. Phần một cần thử xem tháng có 30 hay 31 ngày. Tháng hai là tháng ngoại lệ. Trong phần hai chỉ có hai tháng có 30 ngày là tháng 9 và tháng 11, điều phân chia này làm đơn giản chương trình. Đồng thời cũng tính đến năm nhuận.

```
/* CT86 Chuong trinh lich the ky */
```

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
#define N0 "CHU NHAT"
```

```
#define N1 "Thu Hai"
```

```

#define N2 "Thu Ba"
#define N3 "Thu Tu"
#define N4 "Thu Nam"
#define N5 "Thu Sau"
#define N6 "Thu Bay"
int ngaytuan; /* 1..6 */
int ngay; /* 1..31 */
int thang; /* 1..12 */
int nam; /* 1..3000 */
int theky; /* 0..30 */
int NgayThu(int ,int ,int ,int );
void InDate(void);
int NamNhuan(int );

void main()
{
clrscr();
printf("\n\t\t\tLICH THE KY DEN NAM 3000");
printf("\nHay cho nam can lap LICH");
printf("\nNAM : ");
scanf("%d",&nam);
printf("\n\t\t\tNAM : %6d\n\n",nam);
printf("\n Thang Gieng Thang Hai ");
printf(" Thang Ba Thang Tu ");
printf(" Thang Nam Thang Sau\n");
for (ngay=1;ngay<=31;ngay++)
{
    for (thang=1;thang<=12;thang++)

```

```

if ((thang%2)==0)
    if (thang==2)
        if (ngay>28)
            if ((ngay==29) && (NamNhuan(nam)))
                InDate;
            else printf(" ");
        else InDate;
    else if (ngay==31) printf(" ");
    else InDate;
else InDate;
printf("\n");
}

getch();
printf("\n");
printf("\n Thang Bay Thang Tam ");
printf(" Thang Chin Thang Muoi ");
printf(" Thang Mot Thang Chap\n");
for (ngay=1;ngay<=31;ngay++)
{
    for (thang=7;thang<=12;thang++)
        if ((thang==9) || (thang==11)) && (ngay==31)
            printf(" ");
        else InDate;
    printf("\n");
}
getch();
}

```

```

int NgayThu(int ng,int th,int n,int ngthu)
{
    int nb,an;
    if (th > 2) th = th-2;
    else
    {
        th = th + 10;
        n = n - 1;
    }
    an = n%100;
    theky = (int)(n/100);
    nb = (int)((13*th-1)/5)+(int)(an/4)+(int)(theky/4);
    ngthu =((nb+an+ng-2*theky)%7);
    return(ngthu);
}

void InDate(void)
{
    NgayThu(ngay,thang,nam,ngaytuan);
    switch (ngaytuan)
    {
        case 0 : printf(" %3d %9s",ngay,N0);
                    break;
        case 1 : printf(" %3d %9s",ngay,N1);
                    break;
        case 2 : printf(" %3d %9s",ngay,N2);
                    break;
        case 3 : printf(" %3d %9s",ngay,N3);
                    break;
    }
}

```

```

    case 4 : printf(" %3d %9s",ngay,N4);
               break;
    case 5 : printf(" %3d %9s",ngay,N5);
               break;
    case 6 : printf(" %3d %9s",ngay,N6);
               break;
}
getch();
}
int NamNhuan(int n)
{
if ((n%4)==0) return(1);
else return(0);
}

```

Bài số 87 Xây dựng hàm để kiểm tra dữ liệu vào có phải là số nguyên hay không.

Bài giải Ta xây dựng 3 hàm kiểm tra việc vào dữ liệu :

*void VaoSo1(int * p)* có đối số là kiểu nguyên, trả về một void.
int VaoSo2(void) có đối số kiểu void, trả về một kiểu nguyên.
*int VaoSo3(char * thongbao, int * p, int sai_max)* có điểm số lần vào sai cho phép.

/ CT87 Ham de quy - kiem tra so lieu vao la so nguyen */*

#include <stdio.h>

#define max 10 / so dai nhat co 10 chu so */*

*void VaoSo1(int *);*

int VaoSo2(void);

*int VaoSo3(char *,int *,int);*

```

main()
{
int n;
const solanvaosai = 3; /* Cho function VaoSo3
Neu vao sai > 3 lan thi stop chuong trinh - cho VaoSo3 */
printf("\nCho mot so (kieu 1) : ");
VaoSo1(&n);
printf("\nSo vao dung la : %d\n",n);
getch();
printf("\nCho mot so (kieu 2) : ");
n = VaoSo2();
printf("\nSo vao dung la : %d\n",n);
getch();

/* Vao kieu thu 3 */
n = 0 ; /* Khoi tao lai n */
if (VaoSo3("Cho mot so (kieu 3) : ",&n,solanvaosai) != -1)
    printf("\nSo vao dung la : %d\n",n);
else printf("Vao sai qua tam ba ban - Stop");
getch();
return;
}

void VaoSo1(int *p)
{
int kiemtra;
char conso[max+1]; /* Dung cho lenh fgets ,
                     +1 de tinh den dau ket thuc dong \0 */
fgets(conso,max,stdin);

```

```

kiemtra = sscanf(conso,"%d",p);
if (!kiemtra)
{
    printf("\nVao sai ! Moi vao lai : ");
    VaoSo1(p);
}
int VaoSo2(void)
{
int kiemtra,p;
char conso[max+1]; /* Dung cho lenh fgets
                     +1 de tinh den dau ket thuc dong \0 */
fgets(conso,max,stdin);
kiemtra = sscanf(conso,"%d",&p);
if (!kiemtra)
{
    printf("\nVao sai ! Moi vao lai : ");
    p = VaoSo2();
}
return(p);
}
int VaoSo3(char *thongbao,int *p,int sai_max)
/* sai_max - so lan vao sai cho phep */
{
int kiemtra;
char conso[max+1]; /* Dung cho lenh fgets,
                     +1 de tinh den dau ket thuc dong \0 */
printf("%s",thongbao);

```

```

fgets(conso,max,stdin);
kiemtra = sscanf(conso,"%d",&p);
if (!kiemtra)
    if (--sai_max)
        return(VaoSo3("Vao sai ! Moi vao lai : ",p,sai_max));
    else return(-1);
else return(0);
}

```

Kết quả CT87 như sau :

*Cho mot so (kieu 1) : 15
So vao dung la : 15
Cho mot so (kieu 2) : 25.75
So vao dung la 25
Cho mot so (kieu 3) : a5
Vao sai ! Moi vao lai :*

Bài số 88 Lập trình một tờ hóa đơn

Bài giải Một tờ hóa đơn thường có các mục sau đây :

- Số thứ tự
- Tên loại hàng
- Số lượng hàng
- Đơn giá
- Giá tiền loại hàng
- Tổng giá tiền của hóa đơn.

Chương trình CT88 gồm việc vào dữ liệu dạng mảng chứa tên loại hàng, đơn giá, số lượng và cần thực hiện tính tiền từng loại hàng và tổng tiền của hóa đơn theo thuật toán tính tổng một dãy số (CT23) và in kết quả.

```
/* CT88 Chuong trinh lap hoa don */
#include <stdio.h>
#include <conio.h>
#include <ctype.h>
#define MAX 50 /* so chung loai hang */

void main()
{
    int i,n;
    char OK,Hang[11]; /* Ten hang so ky tu dai nhat la 11 */
    float Dongia[MAX],Soluong[MAX],Tien[MAX];
    float Tongtien;
    clrscr();
    i = 1;
    OK = 'C';           /* Dung de xac dinh so chung loai hang */
    do
    {
        printf("\nTen loai Hang[%d] : ",i);scanf("%s",Hang[i]);
        printf("\n Don gia[%d] : ",i);scanf("%f",&Dongia[i]);
        printf("\n So luong[%d] : ",i);scanf("%f",&Soluong[i]);
        i++;
        printf("\nCo vao hang khac nua khong (c/k) : ");
        getchar();
    }
```

```

if (toupper(getchar())=='C')
    OK = 'C';
else
    OK = 'K';
}
while (OK=='C');
n = i;
printf("\n\n CHUONG TRINH LAP HOA DON ");
printf("\n\n----- HOA DON ----- ");
printf("\n * No * Loai hang * So luong * Don gia USD * Tien
USD **");
i = 1;
Tongtien = 0;
while ( i < n)
{
    Tien[i] = Dongia[i]*Soluong[i];
    Tongtien = Tongtien + Tien[i];
    printf("\n %3d %8s %15.2f %15.2f %15.2f",i, Hang[i], Soluong[i],
          Dongia[i], Tien[i]);
    i++;
}
printf("\n\n Tong so tien : %25.2f ",Tongtien);
getch();
return;
}

```

Theo CT88 ta có tờ hóa đơn như sau :

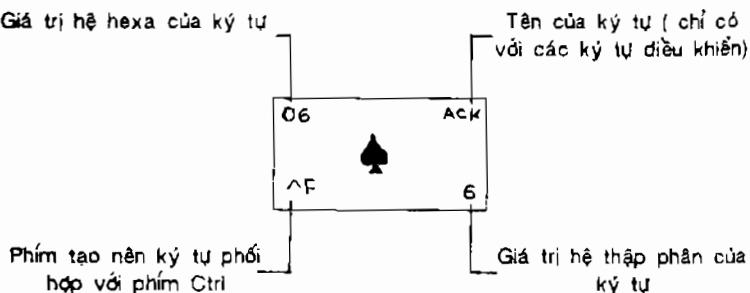
---HOA DON---

* No	* Loai hang	* So luong	* Don gia USD	* Tien USD
1	moto	2.00	1540.00	3080.00
2	radio	3.00	150.00	450.00
3	tivi	1.00	420.00	420.00
Tong so tien		:	3950.00	

Bài số 89 Lập trình in các kiểu kẻ khung ký tự đồ họa mã ASCII.

Bài giải

256 ký tự của mã ASCII được biểu diễn như sau :



Bằng cách bố trí các giá trị hệ hexa của các ký tự theo vị trí toạ độ màn hình (ở chế độ văn bản) ta có thể tạo nên những kiểu in theo ý muốn.

Các chương trình soạn thảo văn bản tiếng Việt như BKED, Vietstar cũng được xây dựng theo nguyên lý này.

```
/* CT89 Cac kieu ke khung dung ky tu do hoa ma ASCII */  
#include <stdio.h>  
#include <conio.h>
```



```

printf("\xBA");
printf("\xBA");
for (i=1;i<79;i++)
    printf(" ");
printf("\xBA");
printf("\xC8"); /* Goc trai duoi N 201 "É" = C8 */
for (i=1;i<79;i++)
    printf("\xCD");
printf("\xBC"); /* Goc phai duoi N 188 "½" = BC */

/* Ke khung don */
printf("\n\xDA");
for (i=1;i<79;i++)
    printf("\xC4");
printf("\xBF");
printf("\xB3");
for (i=1;i<79;i++)
    printf(" ");
printf("\xB3");
printf("\xB3");
for (i=1;i<79;i++)
    printf(" ");
printf("\xB3");
printf("\xC0");
for (i=1;i<79;i++)
    printf("\xC4");
printf("\xD9");
printf("\n");

```

```

/* In bang ma ASCII */
printf("\nIn lan dau\n");
getch();
i=0;
{
    printf("\n%d = \x00 %d = \x01 %d = \x02 %d = \x03 %d = \x04
%d = \x05 %d = \x06 %d = \x07 %d = \x08 %d = \x09 %d =\x0A",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
    printf("\n%d = \x0B %d = \x0C %d = \x0D %d = \x0E %d=\x0F
%d = \x10 %d = \x11 %d = \x12 %d = \x13 %d = \x14 %d =\x15",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
    printf("\n%d = \x16 %d = \x17 %d = \x18 %d = \x19 %d=\x1A
%d = \x1B %d = \x1C %d = \x17 %d = \x18 %d = \x19 %d=\x1A",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
    printf("\n%d = \x1B %d = \x1C %d = \x1D %d = \x1E %d=\x1F
%d = \x20 %d = \x21 %d = \x22 %d = \x23 %d = \x24 %d=\x25",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
    printf("\n%d = \x26 %d = \x27 %d = \x28 %d = \x29 %d =\x2A
%d = \x2B %d = \x2C %d = \x2D %d =\x2E %d =\x2F %d =\x30",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
    printf("\n%d = \x31 %d = \x31 %d = \x33 %d = \x34 %d = \x35
%d = \x36 %d = \x37 %d = \x38 %d = \x39 %d = \x3A %d =\x3B",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
    printf("\n%d = \x3C %d = \x3D %d = \x3E %d = \x3F %d =
\x40 %d = \x41%d =\x42 %d =\x43 %d =\x44 %d =\x45 %d=\x46",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
    getch();
    printf("\n%d =\x47 %d = \x48 %d = \x49 %d =\x4A %d =\x4B
%d =\x4D %d =\x4E %d =\x4F %d =\x50 %d =\x51 %d=\x52",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);
}

```

```

printf("\n%d = \x53 %d = \x54 %d = \x55 %d = \x56 %d = \x57
%d = \x58 %d = \x59 %d = \x5A %d = \x5B %d = \x5C %d = \x5D",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \x5E %d = \x5F %d = \x60 %d = \x61 %d = \x62
%d = \x63 %d = \x64 %d = \x65 %d = \x66 %d = \x67 %d = \x68",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \x69 %d = \x6A %d = \x6B %d = \x6D %d = \x6E
%d = \x6F %d = \x70 %d = \x71 %d = \x72 %d = \x73 %d = \x74",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \x75 %d = \x76 %d = \x77 %d = \x78 %d = \x79
%d = \x7A %d = \x7B %d = \x7C %d = \x7D %d = \x7E %d = \x7F",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \x80 %d = \x81 %d = \x82 %d = \x83 %d = \x84
%d = \x85 %d = \x86 %d = \x87 %d = \x88 %d = \x89 %d = \x8A",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

getch();

printf("\n%d = \x8B %d = \x8C %d = \x8D %d = \x8E %d = \x8F
%d = \x90 %d = \x91 %d = \x92 %d = \x93 %d = \x94 %d = \x95",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \x96 %d = \x97 %d = \x98 %d = \x99 %d = \x9A
%d = \x9B %d = \x9C %d = \x9D %d = \x9E %d = \x9F %d = \xA0",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \xA1 %d = \xA2 %d = \xA3 %d = \xA4 %d =
\xA5 %d = \xA6 %d = \xA7 %d = \xA8 %d = \xA9 %d = \xAA",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \xAB %d = \xAC %d = \xAD %d = \xAE %d = \xAF
%d = \xB0 %d = \xB1 %d = \xB2 %d = \xB3 %d = \xB4 %d = \xB5",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \xB6 %d = \xB7 %d = \xB8 %d = \xB9 %d = \xBA
%d = \xBB %d = \xBC %d = \xBD %d = \xBE %d = \xBF %d = \xC0",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

```

```

printf("\n%d = \xC1 %d = \xC2 %d = \xC3 %d =\xC4 %d =\xC5
%d = \xC6 %d = \xC7 %d = \xC8 %d = \xC9 %d = \xCA",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

getch();

printf("\n%d =\xCB %d =\xCC %d =\xCD %d =\xCE %d =\xCF
%d = \xD0 %d =\xD1 %d =\xD2 %d = \xD3 %d = \xD4 %d=\xD5",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \xD6 %d = \xD7 %d =\xD8 %d =\xD9 %d =\xDA
%d =\xDB %d =\xDC %d =\xDD %d =\xDE %d =\xDF %d =\xE0",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \xE1 %d = \xE2 %d = \xE3 %d =\xE4 %d =\xE5
%d = \xE6 %d = \xE7 %d = \xE8 %d = \xE9 %d = \xEA",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \xEB %d =\xEC %d = \xED %d =\xEF %d =\xF0
%d = \xF1 %d = \xF2 %d = \xF3 %d = \xF4 %d =\xF5 %d =\xF6",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

printf("\n%d = \xF7 %d = \xF8 %d = \xF9 %d =\xFA %d =\xFB
%d = \xFC %d = \xFD %d = \xFE %d = \xFF",
i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1,i+=1);

}

getch();

printf("\nIn lan hai\n");

printf("\n\n\x00 \x01 \x02 \x03 \x04 \x05 \x06 \x07 \x08 \x09
\x0A");

printf("\n\n\x0B \x0C \x0D \x0E \x0F \x10 \x11 \x12 \x13 \x14
\x15");

printf("\n\n\x16 \x17 \x18 \x19 \x1A \x1B \x1C \x17 \x18 \x19
\x1A");

printf("\n\n\x1B \x1C \x1D \x1E \x1F \x20 \x21 \x22 \x23 \x24
\x25");

```

```

printf("\n\n\x26 \x27 \x28 \x2A \x2C \x2D \x2E \x2F
\x30");

printf("\n\n\x31 \x31 \x33 \x34 \x35 \x36 \x37 \x38 \x39 \x3A
\x3B");

getch();

printf("\n\n\x3C \x3D \x3E \x3F \x40 \x41 \x42 \x43 \x44 \x45
\x46");

printf("\n\n\x47 \x48 \x49 \x4A \x4B \x4D \x4E \x4F \x50 \x51
\x52");

printf("\n\n\x53 \x54 \x55 \x56 \x57 \x58 \x59 \x5A \x5B \x5C
\x5D");

printf("\n\n\x5E \x5F \x60 \x61 \x62 \x63 \x64 \x65 \x66 \x67
\x68");

printf("\n\n\x69 \x6A \x6B \x6D \x6E \x6F \x70 \x71 \x72 \x73
\x74");

printf("\n\n\x75 \x76 \x77 \x78 \x79 \x7A \x7B \x7C \x7D \x7E
\x7F");

printf("\n\n\x80 \x81 \x82 \x83 \x84 \x85 \x86 \x87 \x88 \x89
\x8A");

getch();

printf("\n\n\x8B \x8C \x8D \x8E \x8F \x90 \x91 \x92 \x93 \x94
\x95");

printf("\n\n\x96 \x97 \x98 \x99 \x9A \x9B \x9C \x9D \x9E \x9F
\xA0");

printf("\n\n\xA0 \xA1 \xA2 \xA3 \xA4 \xA5 \xA6 \xA7 \xA8 \xA9
\xAA");

printf("\n\n\xAB \xAC \xAD \xAE \xAF \xB0 \xB1 \xB2 \xB3 \xB4
\xB5");

printf("\n\n\xB6 \xB7 \xB8 \xB9 \xBA \xBB \xBC \xBD \xBE \xBF
\xC0");

```

```

printf("\n\n\xC0 \xC1 \xC2 \xC3 \xC4 \xC5 \xC6 \xC7 \xC8 \xC9
\xCA");

printf("\n\n\xCB \xCC \xCD \xCE \xCF \xD0 \xD1 \xD2 \xD3 \xD4
\xD5");

getch();

printf("\n\n\xD6 \xD7 \xD8 \xD9 \xDA \xDB \xDC \xDD \xDE
\xDF \xE0");

printf("\n\n\xE0 \xE1 \xE2 \xE3 \xE4 \xE5 \xE6 \xE7 \xE8 \xE9
\xEA");

printf("\n\n\xEB \xEC \xED \xEF \xF0 \xF1 \xF2 \xF3 \xF4 \xF5
\xF6");

printf("\n\n\xF7 \xF8 \xF9 \xFA \xFB \xFC \xFD \xFE \xFF");
getch();

for (i=0;i<=256;i++)
{
    printf("%d = %c ",i,a[i]);
}

/*printf("\nIn lan ba\n");
j = 0x00;
for (k=0x00;k<=0xFF;k++,j++)
{
    printf("%d = %x",k,*j);
    print(" ");
} */

getch();
}

```

Bài số 90 Lập trình vào_ ra một dữ liệu kiểu cấu trúc.

Bài giải Xây dựng một cấu trúc có trường là một char c và x, y là hai số nguyên là tọa độ của điểm biểu diễn dạng :

```
struct diem
```

```
{  
    char c;  
    int x, y;  
};
```

Xây dựng hai hàm void HienThi1 truy nhập kiểu thông thường và void HienThi2 truy nhập kiểu con trỏ và một cấu trúc để đảo ngược tọa độ x, y của điểm biểu diễn.

```
/* CT90 Kieu cau truc - Vao ra mot du lieu struct */  
  
#include <stdio.h>  
#include <conio.h>  
  
struct diem  
{ char c;  
    int x,y;  
};  
  
void HienThi1(struct diem p); /* Bien thong thuong */  
void HienThi2(struct diem *); /* Bien kieu con tro */  
struct diem Dao(struct diem *); /* Ham dao toa do x,y */  
  
main()  
{  
    struct diem s1,s11,s22;  
    struct diem s2={'B',333,555}; /* Khoi tao bien cau truc s2 */  
    clrscr();
```

```

printf("\n%25cKieu struct cua ngon ngu C."); /* Lui tu dau dong
25 cot */
s1.c = 'A'; /* Gan cac gia tri cho bien cau truc s1 */
s1.x = 111;
s1.y = 222;
HienThi1(s1);
getch();
HienThi2(&s2);
getch();
printf("\nDao nguoc toa do cac diem da cho");
s11 = Dao(&s1); /* Dao nguoc toa do s1 */
HienThi2(&s11); /* Vi ham Dao cho ket qua kieu con tro ,
nen khi hien thi phai dung kieu con tro */
printf("Diem goc s1 = %c %d %d\n",s1.c,s1.x,s1.y);
printf("Diem dao s11 = %c %d %d\n",s11.c,s11.x,s11.y);
getch();
s22 = Dao(&s2);
HienThi2(&s22);
printf("Diem goc s2 = %c %d %d\n",s2.c,s2.x,s2.y);
printf("Diem dao s22 = %c %d %d\n",s22.c,s22.x,s22.y);
getch();
return;
}

void HienThi1(struct diem p) /* Truy nhap kieu thong thuong */
{
    printf("\nDiem %c co toa do la %d %d\n",p.c,p.x,p.y);
}

```

```

void HienThi2(struct diem *diachip) /* Truy nhap kieu con trc */
{
    printf("\nDiem %c co toa do la %d %d\n", diachip->c, diachip->x,
diachip->y);
}

struct diem Dao(struct diem *diachiCA) /* Ham dao nguoc toa
do */
{
    struct diem CA;           /* Diem co gia tri nghich dao */
    CA.c = diachiCA->c;
    CA.x = - diachiCA->x;
    CA.y = - (*diachiCA).y;   /* Thay diachiCA.y = (*diachiCA).y */
    return CA;    /* Gia tri dua ra cua ham : viet return CA hoac
return(CA) */
}

```

Ta được kết quả chương trình CT90 như sau:

Kieu struct cua ngon ngu C

Diem A co toa do la 111 222

Diem B co toa do la 333 555

Dao nguoc toa do cac diem da cho

Diem A co toa do la -111 -222

Diem goc s1 = A 111 222

Diem dao s11 = A -111 -222

Diem B co toa do la 333 555

Diem goc s2 = B 333 555

Diem dao s22 = B -333 -555

Bài số 91 Chương trình tháp Hà nội

Bài giải: Chương trình tháp Hà nội là một ví dụ điển hình của thuật toán đệ quy của hàm, nghĩa là một hàm có thể gọi chính nó. Việc đệ quy cần sử dụng bộ nhớ kiểu xếp chồng LIFO (Last In, First Out stack) để chứa các kết quả trung gian do đó cần thận trọng khi kết thúc quá trình đệ quy.

Bài toán tháp Hà nội phát biểu như sau :

Cho ba chiếc cọc đánh số 1, 2, 3 và n chiếc đĩa có kích thước khác nhau. Ban đầu các đĩa được xếp trên cọc 1 với thứ tự đường kính giảm dần. Cần chuyển n đĩa từ cọc số 1 sang cọc số 3 theo nguyên tắc :

- Mỗi lần chỉ được chuyển một đĩa.
- Mỗi đĩa có thể được chuyển từ một cọc này sang cọc **khác** bất kỳ.
- Không được đặt một đĩa trên đĩa khác có đường kính nhỏ hơn.

Với $n = 1$ thuật toán rất đơn giản, chỉ cần chuyển duy nhất một đĩa từ cọc 1 sang cọc số 3.

Khi $n > 1$ ta cần sử dụng cọc số 2 để chuẩn bị bước trung gian. Có thể phân tích bước trung gian này thành 2 thao tác đơn giản hơn :

- Chuyển $n - 1$ đĩa phía trên của cọc số 1 sang cọc số 2, trong giai đoạn này có thể sử dụng cọc số 3 như cọc trung gian.
- Chuyển $n - 1$ đĩa từ cọc 2 sang cọc số 3, khi đó có thể sử dụng cọc 1 như cọc trung gian.

Điều đó dẫn tới việc xây dựng một hàm đệ quy có các đối số sau đây:

- Số lượng đĩa cần chuyển
- Số cọc xuất phát
- Số cọc chuyển tới
- Số cọc trung gian

```

/* CT91 Chuong trinh thap Ha noi */
#include<stdio.h>
void main()
{
void hanoi (int,int,int);
int nd;          /* tong so dia */
printf("\n Co bao nhieu dia ? ");scanf("%d",&nd);
hanoi( nd,1,3,2);
}

/* ham de quy giai bai toan thap Ha noi */
void hanoi (int n,int xuat , int toi,int trung)
    /* n : so dia can chuyen */
    /* xuat : thap xuat phat */
    /* toi : thap can toi */
    /* trung : thap trung gian */
{
if (n > 0)
{
    hanoi( n-1, xuat,trung,toi);
    printf(" Chuyen dich tu coc %d toi coc %d\n",xuat,toi);
    hanoi (n-1,trung,toi,xuat);
}
getch();
return;
}

```

Kết quả CT91 như sau :

Co bao nhieu dia ? 3

chuyen dich	tu coc	1	toi coc	3
		1		2
		3		2
		1		3
		2		1
		2		3
		1		3

Chú ý khi n = 64 cần thời gian chuyển mất tới 58 tiếng !

Bài số 92 Mảng một chiều kiểu cấu trúc có các điểm là tọa độ của một đường cong.

Bài giải Ta xác định cấu trúc có tên là điểm tương tự CT90 với các tọa độ x, y của đường cong được nhập từ bàn phím và cấu trúc mảng các điểm và hiển thị tọa độ của các điểm.

```
/* CT92 Mang mot chieu kieu cau truc -
   so lieu cac diem cua mot duong cong */
#include <stdio.h>
#include <conio.h>
#define n 3           /* So diem cua mot duong cong */
#define maxline 128 /* Mot dong du lieu nhieu nhat 128 ky tu */
struct diem
{
    char c;
    int x,y;
};
struct diem curve[n];          /* mang cac diem */
void HienThi1(struct diem [],int); /* Kieu struct */
void HienThi(char [],int [],int [],int); /* Kieu don gian */
```

```

main()
{
int i;
char line[maxline]; /* mang cac ky tu du lieu cua 1 diem */
/* cac bien cho kieu don gian */
char Ten[n];
int xx[n],yy[n];
/* Chuong trinh voi struct */
clrscr();
printf("\nCho cac diem cua duong cong-Bien kieu struct");
for (i=1;i<=n;i++)
{
printf("\nTen (1 chu cai) va toa do cua diem %d :",i);
gets(line);
sscanf(line,"%c %d %d",&curve[i].c,&curve[i].x,&curve[i].y);
}
HienThi1(curve,n);
getch();
/* Chuong trinh kieu don gian */
clrscr();
printf("\nCho cac diem cua duong cong-Kieu don gian");
for (i=1;i<=n;i++)
{
printf("\nTen (1 chu cai) va toa do cua diem %d :",i);
gets(line);
sscanf(line,"%c %d %d",&Ten[i],&xx[i],&yy[i]);
}
HienThi(Ten,xx,yy,n);

```

```

getch();
return;
}

void HienThi1(struct diem curve[],int a)
{
int i;
for (i=1 ;i<=a ;i++)
    printf("\nDiem %c toa do %d %d\n",curve[i].c,curve[i].x,
curve[i].y);
}

/* Ham cho kieu don gian */
void HienThi(char o[],int p[],int q[],int a)
{
int i;
for (i=1;i<=a;i++)
    printf("\nDiem %c co toa do la %d %d\n",o[i],p[i],q[i]);
}

```

Kết quả chương trình CT92 như sau :

*Cho cac diem cua duong cong - Bien kieu struc
Ten (1 chu cai) va toa do cua diem 1 : a 12
Ten (1 chu cai) va toa do cua diem 2 : b 56
Ten (1 chu cai) va toa do cua diem 3 : c 25
Diem a co toa do 12 0
Diem b co toa do 56 0
Diem c co toa do 25 0*

Bài số 93 Xây dựng một cấu trúc nhân sự để nhập vào danh sách:

- Họ tên
- Ngày vào cơ quan
- Ngày ra khỏi cơ quan

Bài giải Xây dựng hai cấu trúc có tên là người và date có trường là :

- Họ tên
- Ngày vào cơ quan
- Ngày ra khỏi cơ quan

và một hàm có tên là VaoDS để vào danh sách từng người

```
/* CT93 Bang kieu cau truc -danh sach tung nguoi */
#include <stdio.h>
#include <conio.h>
#define maxTen 40      /* Ten dai nhat la 40 ky tu */
struct date
{
    int ngay,int thang,int nam;
};

struct nguoi
{
    char Ten[maxTen];
    struct date ngayvao; /* ngay vao co quan */
    struct date ngayra; /* ngay ra khoi co quan */
};

void VaoDS(struct nguoi *ds);
```

```

main()
{
    struct nguoi nhvat;
    clrscr();
    VaoDS(&nhvat);
    printf("\nHo ten : %s \nNgay lam : %d%d%d \nNgay thoi viec
%d%d%d",
           nhvat.Ten,
           nhvat.ngayvao.ngay,nhvat.ngayvao.thang,nhvat.ngayvao.nam,
           nhvat.ngayra.ngay,nhvat.ngayra.thang,nhvat.ngayra.nam);
    getch();
    return;
}

void VaoDS(struct nguoi *ds)
{
    char OK;          /* Bien dung cho viec hoi dap */
    printf("\nCho ten: ");
    gets(ds->Ten);   /* Chu y khong kiem tra do dai Ten */
    printf("\nNgay vao co quan : ");
    scanf("%d %d %d",&ds->ngayvao.ngay,
          &ds->ngayvao.thang,
          &ds->ngayvao.nam);
    printf("\nNgay vao = ngay thoi viec ? y/n : ");
    getchar();         /* De xuong dong */
    OK = getchar();
    if (OK == 'y') ds->ngayra = ds->ngayvao;
    else
    {

```

```

printf("\nNgay thoi viec : ") ;
scanf("%d %d %d",&ds->ngayra.ngay,
      &ds->ngayra.thang,
      &ds->ngayra.nam) ;
}

}

```

Kết quả của chương trình CT93 như sau :

Cho tên : Le thu Ha

Ngay vao co quan : 12 5 1990

Ngay vao = ngay thoi viec ? y/n : n

Ngay thoi viec : 14 7 1995

Ho ten : Le thu Ha

Ngay lam : 12 5 1990

Ngay thoi viec : 14 7 1995

Bài số 94 Lập danh sách nhân sự.

Bài giải Để lập cấu trúc danh sách nhân sự dùng hai cấu trúc người và date tương tự như CT93 ở đây có thêm một số người trong danh sách và do đó tạo nên một bảng danh sách đầy đủ.

```

/* CT94 Bang kieu cau truc - Danh sach nhan su */
#include <stdio.h>
#include <conio.h>
#define maxTen 40 /* Ten dai nhat la 40 ky tu */
#define max 50     /* So nguoi nhieu nhat trong danh sach */
struct date
{

```

```

int ngay,int thang,int nam;
};

struct nguoi
{
    char Ten[maxTen];
    struct date ngayvao; /* ngay vao co quan */
    struct date ngayra; /* ngay ra khoi co quan */
};

void VaoDS(struct nguoi *ds);

main()
{
    int i,n;
    struct nguoi nhvat[max];
    clrscr();
    printf("\nCho so nguoi trong danh sach n = ");
    scanf("%d",&n);
    for (i=1;i<=n;i++)
    {
        printf("\nNguoi thu %d : ",i);
        VaoDS(&nhvat[i]);
    }
    clrscr();
    for (i=1;i<=n;i++)
    {
        printf("\nNguoi thu %d",i);
        printf("\nHo ten : %s \nNgay vao lam :%d%d%d\nNgay thoigiec
d%d%d",
        nhvat[i].Ten,

```

```

nhvat[i].ngayvao.ngay,nhvat[i].ngayvao.thang,nhvat[i].ngayvao.nam,
nhvat[i].ngayra.ngay,nhvat[i].ngayra.thang,nhvat[i].ngayra.nam);
}
getch();
return;
}

void VaoDS(struct nguoi *ds)
{
char OK;           /* Bien dung cho viec hoi dap */
int i ;
printf("\nCho ten nguoi : ");
scanf("%s",ds->Ten);
/* gets(ds->Ten);    /* Chu y khong kiem tra do dai Ten */
   /* Lenh nay thuong hay bi nhay qua */
printf("\nNgay vao co quan : ");
scanf("%d %d %d",&ds->ngayvao.ngay,
      &ds->ngayvao.thang,
      &ds->ngayvao.nam) ;
printf("\nNgay vao = ngay bi duoi viec ? y/n : ") ;
getchar();        /* De xuong dong */
OK = getchar();
if (OK == 'y') ds->ngayra = ds->ngayvao;
else
{
  printf("\nNgay bi thoi viec : ");
  scanf("%d %d %d",&ds->ngayra.ngay,
        &ds->ngayra.thang,
        &ds->ngayra.nam);
}

```

Kết quả chương trình CT94 như sau :

Cho so nguoi trong danh sach 2

Nguoi thu 1

Ho ten : Le thu Thuy

Ngay vao lam : 15 8 1990

Ngay thoi viec : 0

Nguoi thu 2

Ho ten : Ho van Tam

Ngay vao lam : 12 9 1975

Ngay thoi viec : 15 10 1994

Bài số 95 Lập bảng danh sách nhân sự kiểu cấu trúc dòng.

Bài giải Xây dựng hai cấu trúc để tổ chức danh sách nhân sự gồm:

Họ và tên

Ngày sinh

Quê quán

Mức lương

và lập hai hàm để vào danh sách và ra khỏi danh sách nhân sự.

```
/* CT95 Bang kieu cau truc dong - Danh sach nhan su */
```

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
#define chuoi 50 /* Dai nhat la 50 ky tu */
```

```

struct date
{
    int ngay,int thang,int nam;
};

typedef struct phantu
{
    char Ten[chuoi];
    struct date namsinh;
    char que[chuoi];
    int luong;
    struct phantu *tiep; /* kieu struct cap hai la con tro
        tro den phan tu tiep theo */
} t_phantu; /* Kieu struct : phantu,bien struct : t_phantu */

int sott,songuo;
void VaoDS(t_phantu **); /* Bien con tro tro den mot con tro
khac */
void RaDS(t_phantu *);

main()
{
    t_phantu *khoidau; /* Con tro tro den dau danh sach */
    clrscr();
    VaoDS(&khoidau);
    RaDS(khoidau);
    getch();
    return;
}

```

```

void VaoDS(t_phantu **qq)
{
    char tenvao[chuoi], quevao[chuoi];
    t_phantu *hientai; /* De thay doi gia tri cua con tro */
    clrscr();
    *qq = NULL; /* Khoi dau danh sach rong */
    songuoi = 0;
    while (1) /* Vong lap vao danh sach khong han che so luong */
    {
        printf("Cho ten nguoi,khong cho nua thi go Enter\n");
        printf("Ten nguoi : ");
        gets(tenvao);
        if (strlen(tenvao)) /* Neu co vao ten nguoi */
        {
            songuoi += 1;
            hientai = (t_phantu *)malloc(sizeof(t_phantu));
            strcpy(hientai->Ten,tenvao);
            printf("Nam sinh (cac so cach nhau) : ");
            scanf("%d %d %d",&hientai->namsinh.ngay,
                  &hientai->namsinh.thang,
                  &hientai->namsinh.nam);

            printf("Que quan : ");
            gets(quevao);
            gets(quevao);
            strcpy(hientai->que,quevao);
            printf("Luong : ");
            scanf("%d",&hientai->luong).
    }
}

```

Kết quả chương trình CT95 như sau :

Cho ten nguoi, khong cho nua go Enter

***** Danh sach *****

No Ho va ten	Ngay sinh	Que quan	Luong
1 Le thu Thuy	24 12 1979	Bat trang Gia Lam	100000
2 Nguyen van Long	12 6 1990	Pho Hue Ha noi	200000
3 Nguyen dieu Hoa	15 10 1985	Ba vi Ha noi	250000

Tong so nguoi trong danh sach la 3

Bài số 96 Tìm biểu tượng chòm sao theo ngày và tháng sinh.

Bài giải Theo lịch một số nước phương Tây có 12 chòm sao tương ứng với ngày sinh và tháng sinh sau đây :

Capricorne	23 tháng chạp đến 19 tháng giêng
Verseau	20 tháng giêng đến 19 tháng hai
Poisson	20 tháng hai đến 20 tháng ba
Bélier	21 tháng ba đến 19 tháng tư
Taurau	20 tháng tư đến 20 tháng năm
Gémeau	21 tháng năm đến 20 tháng sáu
Cancer	21 tháng sáu đến 21 tháng bảy
Lion	22 tháng bảy đến 22 tháng tám
Vierge	23 tháng tám đến 22 tháng chín
Balance	23 tháng chín đến 22 tháng mười
Scorpion	23 tháng mười đến 21 tháng mười một
Sagittaire	22 tháng mười một đến 22 tháng chạp

Chương trình cần dự tính xác định mối tương quan giữa tên một biểu tượng chòm sao và hai giá trị thời gian ngày sinh và tháng sinh tương ứng. Có thể tiến hành công việc kể trên theo các cách sau :

- Dùng nhiều mảng (ngày, tháng, chòm sao) liên hệ với nhau qua một chỉ số chung.

- Bằng một mảng cấu trúc mỗi phần tử gồm một số ứng với ngày sinh, một tên ứng với tháng sinh và một tên đặc trưng cho biểu tượng chòm sao. Phương án này cho ta mỗi lệnh hệ rõ ràng giữa các dữ liệu ở thời điểm khởi đầu chương trình. Việc tìm chòm sao tương ứng với ngày sinh và tháng sinh tiến hành như sau :

+ Đầu tiên tìm phần tử thuộc mảng cấu trúc có tên tháng ứng với dữ liệu đã cho, nếu không tương ứng sẽ hiển thị bằng dòng " sai tên tháng "

+ Tiếp theo sẽ tìm số ngày nhỏ hơn hoặc bằng ngày của phần tử tìm kiếm X.

Nếu ngày đó trước ngày cuối của biểu tượng ứng với phần tử X điều đó cho ta kết quả mong muốn.

+ Ngược lại ta có thể kết luận được rằng ngày đó sau ngày bắt đầu của biểu tượng ứng với phần tử X, ta chỉ cần xem xét phần tử tiếp theo để tìm phần tử mong muốn. Tuy nhiên nếu X là phần tử cuối của mảng thì tiếp theo nó phải là phần tử đầu của mảng.

- Chương trình CT96 sử dụng cấu trúc s_date trong đó có một số ứng với ngày sinh tên tháng sinh và biểu tượng chòm sao tương ứng. Ta trù tính 10 ký tự cho tên tháng cho phép một string nhỏ hơn hoặc bằng 9 ký tự (Lưu ý ký tự \0 là ký tự kết thúc) và 11 ký tự dành cho tên biểu tượng chòm sao.

Mảng date là 12 phần tử, mỗi phần tử kiểu s-date đã được khởi đầu điều đó cho phép dễ dàng đặt song song mỗi biểu tượng và thời gian cuối của biểu tượng.

Hàm strcmp cho phép so sánh xâu ký tự không phân biệt chữ in và chữ thường

Việc tìm tháng sinh được thực hiện bằng một lệnh :

```
while ( strcmp ( date[i].thang, thang-s ) && i ++ < 11 ){}  
vừa ngắn gọn, vừa cho phép biết ngay việc tìm kiếm có kết quả hay  
không. Cuối lệnh này i bằng 12 khi i < 12, i chỉ phần tử tìm thấy.
```

Việc sử dụng toán tử số học % cho phép giải quyết vấn đề biểu tượng
tiếp theo biểu tượng cuối cùng của mảng.

Ở đây việc tìm kiếm tên tháng được thực hiện bằng thuật toán tìm
kiếm tuần tự theo mảng.

```
/* CT96 Chương trình tìm sao theo ngày sinh */  
#include<stdio.h>  
#include<conio.h>  
#include<string.h>  
void main()  
{  
struct s_date  
{  
    int ngay;  
    char thang[10];  
    char sao[12];  
}; /* Tên 12 chòm sao */  
struct s_date date[12] = {  
    23, "chap ", "Sagittaire",  
    20, "gieng", "Capricorne",  
    20, "hai" , "Verseau" ,  
    21, "ba" , "Poisson" ,  
    20, "tu" , "Belier" ,  
    21, "nam" , "Taureau" ,  
    21, "sau" , "Gemeau" ,
```

```

        22, "bay" , "Cancer" ,
        23, "tam" , "Lion" ,
        23, "chin" , "Vierge" ,
        23, "muoi" , "Balance" ,
        22, "mot" , "Scorpion"
    };

int ngay_s;
char thang_s[10];
int i;
printf("\n Ban cho biet ngay va thang sinh ? :");
scanf(" %d %s",&ngay_s,thang_s);
printf("\n Ngay sinh : %d",ngay_s);
printf("\n Thang sinh : %s",thang_s);

/* Tim kiem va hien thi chom sao tuong ung */
i = 0;
while (strcmp(date[i].thang,thang_s) && i++ <11) {}
if (i < 12)
{
    printf("\n Ban duoc sinh duoi chom sao : ");
    if ( ngay_s >= date[i].ngay) i = (i+1)%12;
    printf("%s", date[i].sao );
}
else printf(" ***** Sai ten thang ***** ");
getch();
return;
}

```

Trong chương trình CT96 nếu cho các dữ liệu ngày và tháng sinh ta sẽ được kết quả như sau :

Ban cho biet ngay va thang sinh ? : 14 muoi

Ngay sinh : 14

Thang sinh : muoi

Ban duoc sinh duoi chom sao : Balance

Bài số 97 Lập chương trình file truy nhập tuần tự.

Bài giải Chương trình cho phép hiển thị lần lượt các danh sách nhân sự như các chương trình CT94, CT95 kèm theo một số thứ tự trong danh sách.

```
/* CT97 Chuong trinh file truy nhap tuan tu */
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#define maxten 30      /* Ten nhat 20 ky tu */
#define maxcon 15      /* So con nhieu nhat */
#define maxtenfile 40  /* Ten file va duong dan max 40 ky tu */
struct nguoi
{
    char ten[maxten+1];/* +1 de tinh den ca ky tu ket thuc \0 */
    int tuoi;
    int socon;
    int tuoicon[maxcon];
} hoson;
void Hienthi(struct nguoi *,int);
```

```

main()
{
char tenfile[maxtenfile+1]; /* +1 de tinh den ca ky tu
                               ket thuc \0 */

FILE *f1;
int i,n;
/* Tao file */
printf("\nCho ten file : ");
gets(tenfile);
/* Hoac dung lenh fgets(tenfile,maxtenfile,stdin) de phong viec
   cho duong dan dai qua so ky tu maxtenfile */
if ((f1=fopen(tenfile,"w"))==NULL)
{
    printf("\nLoi mo file - Chuong trinh ket thuc\n");
    exit(-1);
}

/* Vao so lieu */
printf("Vao so lieu , muon thoii thi den muc Ten an Enter\n");
n = 0; /* Dem so phan tu trong file */
do
{
    printf("Ten : ");
    gets(hoso.ten);
    if (strlen(hoso.ten)==0) break; /* Ket thuc neu ten la rong */
    n++;
    printf("Tuoi : ");
    scanf("%d",&hoso.tuoi);
    printf("So con : ");
}

```

```

scanf("%d",&hososocon);
for (i=0;i<hososocon && i<maxcon;i++)
{
    printf("Troi con thu %d : ",i+1);
    scanf("%d",&hosotuoicon[i]);
}
getchar(); /* Xuong dong trong file sau mot nguoi */
printf("\n");
fwrite(&hososocon,sizeof(hososo),1,f1); /* Ghi vao file */
}

while (1);
fclose(f1);
printf("\nKet thuc viec tao file va ghi du lieu vao.");
printf("\nTrong file nay co %d phan tu (nguoi)",n);
getch();
/* Hien thi noi dung file */
do
{
    printf("\nCho ten file can doc : ");
    gets(tenfile);
    /* Hoac dung lenh fgets(tenfile,maxtenfile,stdin) de phong viec
       cho duong dan dai qua so ky tu maxtenfile */
    if ((f1=fopen(tenfile,"rb"))==0)
        printf("\nKhong tim thay file - Cho lai ten\n");
}
while ( !f1);
n = 1;
while (fread(&hososo,sizeof(hososo),1,f1))

```

```

{
Hienthi(&hoson, n++);
getchar();
}

fclose(f1);
printf("\nDoc xong danh sach trong file .");
/* Them du lieu vao cuoi file */
do
{
printf("\nCho ten file can them du lieu : ");
gets(tenfile);
if ((f1=fopen(tenfile,"ab"))==0) /* Mo de them vao cuoi file */
    printf("\nKhong tim thay file - Cho lai ten\n");
}
while (!f1);
do
{
printf("Ten : ");
gets(hoso.ten);
if (strlen(hoso.ten)==0) break ; /* Ket thuc neu ten la rong */
n++;
printf("Tuoi : ");
scanf("%d",&hososo.tuoi);
printf("So con : ");
scanf("%d",&hososo.socon);
for (i=0 ; i<hososo.socon && i<maxcon ; i++)
{
printf("Tuoi con thu %d : ",i+1);
}
}

```

```

scanf("%d",&hososo[i]);
}

getchar(); /* Xuong dong trong file sau mot nguoi */
printf("\n");
fwrite(&hososo,sizeof(hososo),1,f1); /* Ghi vao file */
}

while (1);
fclose(f1);
printf("\nKet thuc viec ghi them du lieu vao cuoi file .");
getch();
return;
}

void Hienthi(struct nguoi *hososo,int so)
{
int i ;
printf("\nSo ho so : %d\n",so);
printf("Ten : %s\n",hososo->ten);
printf("Tuoi : %d\n",hososo->tuoi);
printf("So con : %d\n",hososo->socon);
for (i=0;i<hososo->socon && i<maxcon;i++)
    printf("Tuoi con thu %d : %d\n",i+1,hososo->tuoicon[i]);
}

```

Kết quả chương trình CT97 như sau :

Cho ten file : Danh sach

Vao so lieu muon them den muc Ten an Enter

Ten : Nguyen Kim Dung

Tuoi : 45

So con : 2

Tuoi con thu nhat : 21

Tuoi con thu hai : 15

Ket thuc viec tao file va ghi du lieu vao

Trong file nguoi co 1 phan tu (nguoi)

Cho ten file can doc : Danh sach

Doc xong danh sach trong file

Cho ten file can them du lieu

Ket thuc ghi len file

Bài số 98 Lập trình tạo file truy nhập tuần tự cho phép tạo, xem, sửa, thêm vào cuối danh sách và cho phép chọn tên danh sách.

Bài giải CT98 là chương trình đầy đủ tạo nên một danh sách hồ sơ cho phép.

- Tạo danh sách
- Hiển thị danh sách
- Sửa danh sách
- Thêm vào danh sách
- Đặt tên danh sách

Các lời giải thích đã nằm trong văn bản chương trình. Ta được kết quả của chương trình CT98 tương tự CT97 nhưng đầy đủ hơn.

```
/* CT98 Chuong trinh file truy nhap tuan tu - :  
   tao,xem,sua,them vao cuoi file,cho phep chon ten file */  
#include <stdio.h>  
#include <conio.h>
```

```

#include <string.h>
#include <stdlib.h>

#define maxten 30      /* Ten dai nhat 30 ky tu */
#define maxcon 15      /* So con nhieu nhat */
#define maxtenfile 40  /* Ten file va duong dan max 40 ky tu */

struct nguoi
{
    char ten[maxten+1]; /* +1 tinh ca ky tu ket thuc \0 */
    int tuoi;
    int socon;
    int tuoicon[maxcon];
} hoso;

char tenfile[maxtenfile+1]; /* +1 tinh ca ky tu ket thuc \0 */
FILE *f1;
int n,   /* so ho so */
ngoai; /* so > so ho so */
long sohososo,vitri; /* vi tri hien tai trong file */
char dong[maxtenfile+1];
void TaoFile(void);
void Hienthi(struct nguoi *,int);
void Sua(struct nguoi *);
void Them(void);

main()
{
    int i;
    /* Mo file */
    printf("\n***** MO FILE *****\n");

```

```

do
{
    printf("\nCho ten file can mo : ");
    gets(tenfile);
    if ((f1=fopen(tenfile,"wt"))==NULL)      /* wt - mo de ghi
                                                file Text */
        printf("\nLoi mo file - Cho lai ten file\n");
}
while ( !f1);

TaoFile();
/* Hien thi noi dung file */
do
{
    printf("\n***** XEM NOI DUNG FILE *****\n");
    printf("\nTen file dang soan : %s\n",tenfile);
    printf("\nCho ten file (neu la file dang soan thi an Enter) : ");
    gets(dong);      /* Cho ten moi */
    if (strlen(dong)) strcpy(tenfile,dong);
    if ((f1=fopen(tenfile,"rt"))==0)
        printf("\nKhong tim thay file - Cho lai ten\n");
}
while ( !f1);

n = 1;
while (fread(&hoson,sizeof(hoso),1,f1))
{
    Hienthi(&hoson,n++);
    getchar();
}

```

```

fclose(f1);
printf("\nDoc xong danh sach trong file.\n\n");
/* Sua du lieu trong file */
do
{
    printf("\n***** SUA CHUA NOI DUNG FILE *****\n");
    printf("\nTen file dang soan : %s\n",tenfile);
    printf("\nCho ten file (neu la file dang soan thi an Enter): ");
    gets(dong); /* Cho ten moi */
    if (strlen(dong)) strcpy(tenfile,dong);
    if ((f1=fopen(tenfile,"r+t"))==0)
        printf("Khong tim thay file tren dia\n");
}
while( !f1);
fseek(f1,0,2);
sohoso = ftell(f1) / sizeof(hoso);
/* Tim va sua ho so */
do
{
    do
    {
        printf("\nCho so ho so can sua (go 0 de stop) : ");
        scanf("%d",&n);
        getchar(); /* De nhay qua ki tu cuoi \n */
        ngoai = n<0 || n>sohoso;
    }
    while (ngoai);
    if (n==0) break; /* Khong sua ra khoi vong lap */
}

```

```

vitri = (n-1)*sizeof(hoso); /* Tinh vi tri cua ho so can sua */
fseek(f1,vitri,0); /* Dinh vi con tro den ho so */
fread(&hoso,sizeof(hoso),1,f1); /* Doc mot ho so vao bo nho */
Hienthi(&hoso,n); /* Hien thi noi dung ho so o bo nho */
Sua(&hoso); /* Sua ho so o bo nho */
fseek(f1,vitri,0);
fwrite(&hoso,sizeof(hoso),1,f1); /* Ghi ho so da sua vao file */
}

while(1);

fclose(f1);

printf("\nDa sua xong\n\n");

/* Xem lai noi dung file da sua */
printf("\n***** XEM LAI NOI DUNG FILE *****\n");
if ((f1=fopen(tenfile,"rt"))!=0)
    n = 1;
while (fread(&hoso,sizeof(hoso),1,f1))
{
    Hienthi(&hoso,n++);
    getchar();
}
fclose(f1);

printf("\nDoc xong danh sach trong file .\n\n");
getch();

/* Them du lieu vao cuoi file */
do
{
    printf("\n***** THEM DU LIEU VAO CUOI FILE *****\n");
    printf("\nTen file dang soan : %s\n",tenfile);

```

```

printf("\nCho ten file (neu la file dang soan thi an Enter) : ");
gets(dong); /* Cho ten moi */
if (strlen(dong)) strcpy(tenfile,dong);
if ((f1=fopen(tenfile,"at"))==0) /* Mo file de them vao cuoi */
    printf("Khong tim thay file tren dia\n");
}
while( !f1);
Them();
/* Hien thi noi dung file */
do
{
printf("\n***** XEM NOI DUNG FILE *****\n");
printf("\nTen file dang soan : %s\n",tenfile);
printf("\nCho ten file (neu la file dang soan thi an Enter) : ");
gets(dong); /* Cho ten moi */
if (strlen(dong)) strcpy(tenfile,dong);
if ((f1=fopen(tenfile,"rt"))==0)
    printf("\nKhong tim thay file - Cho lai ten\n");
}
while ( !f1);
n = 1;
while (fread(&hos0,sizeof(hos0),1,f1))
{
Hienthi(&hos0,n++);
getchar();
}
fclose(f1);
printf("\nDoc xong danh sach trong file.Enter to stop !");

```

```

getch();
return;
}

void TaoFile(void)
{
int i;
printf("Vao so lieu , muon thoig thi den muc Ten an Enter\n");
n = 0; /* Dem so phan tu trong file */
do
{
    printf("Ten    : ");
    gets(hoso.ten);
    if (strlen(hoso.ten)==0) break; /* Ket thuc neu ten la rong */
    n++;
    printf("Tuoi    : ");
    scanf("%d",&hoso.tuoi);
    printf("So con : ");
    scanf("%d",&hoso.socon);
    for (i=0;i<hoso.socon && i< maxcon;i++)
    {
        printf("Tuoi con thu %d : ",i+1);
        scanf("%d",&hoso.tuoicon[i]);
    }
    getchar(); /* Xuong dong trong file sau mot nguoi */
    printf("\n");
    fwrite(&hoso,sizeof(hoso),1,f1); /* Ghi vao file */
}

```

```

while (1);
fclose(f1);
printf("\nKet thuc viec tao file va ghi du lieu vao .");
printf("\nTrong file nay co %d phan tu (nguois,n);
getch();
}

void Hienthi(struct nguois *hosoi,int so)
{
int i;
printf("\nSo ho so : %d\n",so);
printf("Ten      : %s\n",hosoi->ten);
printf("Tuoi      : %d\n",hosoi->tuoi);
printf("So con   : %d\n",hosoi->socon);
for (i=0;i<hosoi->socon && i<maxcon;i++)
    printf("Tuoi con thu %d : %d\n",i+1,hosoi->tuoicon[i]);
}

void Sua(struct nguois *hosoi)
{
int i;
printf("Sua chua so lieu , go Enter neu khong sua\n");
printf("Ten      : ");
gets(dong); /* Cho ten moi */
if (strlen(dong)) strcpy(hosoi->ten,dong);
printf("Tuoi      : ");
gets(dong); /* Cho tuoi moi */
if (strlen(dong)) hosoi->tuoi = atoi(dong);
printf("So con   : ");

```

```

gets(dong); /* Cho so moi */
if (strlen(dong)) hoso->socon = atoi(dong);
for (i=0;i<hoso->socon && i<maxcon;i++)
{
    printf("Tuoi con thu %d : ",i+1);
    gets(dong); /* Cho so moi */
    if (strlen(dong)) hoso->tuoicon[i] = atoi(dong);
}
}

void Them(void)
{
int i;
printf("Vao so lieu , muon thoi thi den muc Ten an Enter\n");
do
{
    printf("Ten      : "); gets(hoso.ten);
    if (strlen(hoso.ten)==0) break; /* Ket thuc neu ten la rong */
    n++;
    printf("Tuoi      : ");
    scanf("%d",&hoso.tuoi);
    printf("So con   : ");
    scanf("%d",&hoso.socon);
    for(i=0;i<hoso.socon && i<maxcon;i++)
    {
        printf("Tuoi con thu %d : ",i+1);
        scanf("%d",&hoso.tuoicon[i]);
    }
}

```

```

getchar(); /* Xuong dong trong file sau mot nguoi */
printf("\n");
fwrite(&hoson,sizeof(hoso),1,f1); /* Ghi vao file */
}
while (1);
fclose(f1);
printf("\nKet thuc viec them du lieu vao cuoi file.");
getch();
}

```

Bài số 99 Lập trình mã hóa morse và giải mã morse một bức điện đánh từ bàn phím.

Bài giải Ta biết mã morse gồm 26 ký tự chữ cái từ A đến Z (giả thiết đánh bằng chữ hoa), 10 con số từ 0 đến 9 và dấu chấm như sau:

BÀNG MÃ MORSE

A	• -	B	- • •	C	- • - •	D	- • •	E	•
F	• • - •	G	- - •	H	• • •	I	• •	J	• - - -
K	- • -	L	• - • •	M	- -	N	- •	O	- - -
P	• - - •	Q	- - - •	R	• - •	S	• • •	T	-
U	• • -	V	• • - -	W	• - -	X	- • - -	Y	- • - - -
Z	- - - •		• - - - -						
O	- - - -	1	• - - - -	2	• - - - -	3	• - - - -	4	• - - - -
5	• - - - -	6	- - - - •	7	- - - - •	8	- - - - •	9	- - - - •

Chương trình cần xác định sự tương ứng giữa các ký tự chữ và con số quen biết với các mã morse. Để làm việc này ta sử dụng một mảng cấu trúc gọi là code. Mỗi phần tử của mảng này gồm có :

- Một ký tự (từ A đến Z, từ 0 đến 9)
- Mã Morse tương ứng với ký tự được biểu thị dưới dạng string các dấu • -

Ta xác định một struct có tên là code gồm một chữ và một pointer trỏ về một string chứa mã Morse tương ứng.

Mảng chỉ chứa 37 phần tử (giá trị NL) kích thước nói chung gồm 5 byte 1 byte cho ký tự, 4 byte dành cho pointer

Quá trình dịch một bức điện bằng chữ sang mã Morse tiến hành :

- Đọc bức điện từ bàn phím mỗi dòng cực đại 128 ký tự (chữ hoa).
- Việc tìm mã Morse tương ứng được thực hiện bằng một lệnh :

```
while ( dong [i]! = mang [j] . chu && j ++ < NL - 1 );
```

Giải mã Morse là quá trình ngược lại nghĩa là cần tìm một ký tự tương ứng với một mã Morse. Để làm việc này chúng ta có thể sử dụng cùng một cấu trúc như đã làm với quá trình mã hóa, vì thế thuật toán tìm kiếm cũng tương tự. Quá trình giải mã Morse cụ thể như sau :

- Đọc bức điện Morse từ bàn phím bằng gets trong mảng cực đại 128 ký tự có tên là dong. Quá trình xử lý sau đây sẽ lặp lại và dựa trên việc sử dụng con trỏ char * Morse :

- Tăng giá trị con trỏ khi có một khoảng cách.
- Nhờ sscanf lấy một string có độ dài lớn nhất là 7 (không có mã Morse nào vượt quá kích thước này).
- Tăng giá trị con trỏ của string đã được đọc.

Việc tìm mã Morse trong mảng được thực hiện chỉ bằng một lệnh :

```
while ( strcmp ( code, mang [ij] . Morse ) && j ++ < NL - 1 );
```

```

/* CT99 Chuong trinh ma hoa va giao ma morse */
#include<stdio.h>
#include<string.h>
#define NL 37 /* so ky tu duoc ma hoa */
#define LG 127 /* chieu dai dong morse */

void main()
{
    struct code
    {
        char chu;
        char * morse;
    };
    struct code mang[NL] =
    {
        'A', ".-", 'B', "-...", 'C', "-.-.", .
        'D', "-..", 'E', ".", 'F', "...", .
        'G', "--.", 'H', "...", 'I', "..", .
        'J', ".---", 'K', "-.-", 'L', ".--", .
        'M', "--", 'N', "-.", 'O', "----", .
        'P', ".--.", 'Q', "---.", 'R', ".~.", .
        'S', "...", 'T', "-", 'U', "...", .
        'V', "...-", 'W', "...", 'X', "-..-", .
        'Y', "-.--", 'Z', "-.-.", ' ', ".-.-.", .
        '0', "----", '1', "----", '2', "...--",
        '3', "...--", '4', "...--", '5', ".....",
        '6', "...--", '7', "...--", '8', "----",
        '9', "----."
    };
    char dong[LG+1]; /* de doc 1 dong tu ban phim */

```

```

char code[7]; /* ma can dich */
int i,j
/* Doc buc dien */
printf("\n Cho noi dung cua buc dien 1 dong max 128 \n");
gets (dong);
printf("\n Day la ban dich ma Morse cua buc dien\n ");
/* Dich buc dien tung chu */
~ for ( i = 0; i < strlen(dong); i++)
{
    j = 0;
    while (dong[i] != mang[j].chu && j++ <NL-1);
    if (j <NL) printf( "%7s", mang[j].morse);
    else printf( "??????");
    if(! ((i+1)%10)) printf("\n"); /* 10 ma Morse 1 dong */
}
/* Doc buc dien Morse can dich */
printf("\n Cho buc dien Morse ( 1 dong max ) :\n ");
gets(dong);
printf("\n Day la noi dung buc dien \n");
i = 0;
while (i < strlen(dong))
{
    .
    .
    while(dong[i] == ' ') i++; /* nhay qua dau cach */
    if (i < strlen(dong))
    {
        sscanf (&dong[i], "%6s",code); /* doc ma (max 6 ky tu */
        i += strlen(code); /* tang con tro trong dong */
        j = 0;             /* tim ma trong mang */

```

```

        while(strcmp(code,mang[j].morse) && j+ + <NL-1) ;
        if (j < NL) printf("%2c",mang[j].chu) ; /* ma tim thay */
        else printf(" ? ") ; /* ma khong thay */
    }
}

```

Kết quả của chương trình CT99 như sau :

Cho noi dung cua buc dien 1 dong max 128

VO DE VE NGAY

Day la ban dich ma morse cua buc dien

..._ _ _ _ _ ?????? _.. .?????? .._ .?????? _.

--' ' - - - -

Cho mot buc dien morse (1 dong max) :

..._ _ _ -

Day la noi dung cua buc dien

VO DE VE NGAY

Bài số 100 Lập trình cho một bản nhạc trên máy vi tính. Ghi bài Tiến quân ca của nhạc sỹ Văn Cao.

Bài giải Một nốt nhạc được đặc trưng bởi cao độ (tần số) và trường độ.

Có thể nhận được hiệu ứng âm thanh bằng việc lập trình bộ thời gian có thể lập trình PIT (Programmable Interval Timer) và bộ ghép nối ngoại vi có thể lập trình PI (Progammable Peripheral Interface).

Bằng việc lập trình PIT ta tác động lên tần số và khoảng kéo dài của âm thanh. Các hàm sau đây quyết định đến đặc tính âm thanh :

void sound (tanso) - để loa phát một âm với tần số cho trước.

void delay (thoigian) - để loa kéo dài âm phát ra trong khoảng thời gian cho trước.

void nosound (void) - để ngừng loa.

Trước hết ta xác định tần số của các nốt nhạc ứng với các octave, ở đây ta sử dụng 4 octave. Trường độ của nốt nhạc được xác định theo độ dài của nốt đen d và xác định độ dài của nốt trắng, nốt tròn, nốt móc đơn, v.v..., theo nốt đen. Cuối cùng gọi hàm bằng not (cao độ, trường độ); theo trật tự của bản nhạc. Dấu lặng nhận được bằng cách phát ra một siêu âm có tần số 30.000 Hz.

Chương trình CT100 ghi bản nhạc Tiến quân ca của nhạc sỹ Văn Cao, đồng thời với hình ảnh lá cờ đỏ sao vàng. Ngôi sao vàng năm cánh nhận được bằng cách sử dụng hàm fillpoly (10, poly10) để tô màu một đa giác có mười đỉnh là các điểm (x1, y1) ... (x10, y10) là các tọa độ của đỉnh ngôi sao. Tiết tấu nhanh, chậm của bản nhạc có thể thay đổi được bằng cách thay đổi giá trị của nốt đen d. Tiết tấu trung bình nhận được khi d = 300.

Chương trình có một số warning do chưa sử dụng hết một số định nghĩa về trường độ của nốt nhạc như các nốt móc kép, móc ba... Để tổng quát chúng tôi vẫn để các định nghĩa này tuy bài Tiến quân ca không dùng đến.

```
/* CT100 Chuong trinh sang tac nhac Bai Quoc ca */
#include<stdio.h>
#include<stdlib.h>
#include<graphics.h>
#include<dos.h>
/* Xac dinh cao do cua cac not nhac */
#define do1 66      /* do octave 1 */
#define dod1 70     /* do thang */
```

```
#define rel 73      /* re */
#define red1 78      /* re thang */
#define mil 82       /* mi */
#define fa1 86        /* fa */
#define fad1 91      /* fa thang */
#define sol1 96       /* sol */
#define sold1 102     /* sol thang */
#define la1 108       /* la */
#define lad1 115      /* la thang */
#define sil 122       /* si */
#define do2 130       /* do octave 2 */
#define dod2 139      /* do thang */
#define re2 148       /* re */
#define red2 156      /* re thang */
#define mi2 164       /* mi */
#define fa2 176       /* fa */
#define fad2 188      /* fa thang */
#define sol2 196      /* sol */
#define sold2 209     /* sol thang */
#define la2 220       /* la */
#define lad2 233      /* la thang */
#define si2 247       /* si */
#define do3 264       /* do octave 3 */
#define dod3 281      /* do thang */
#define re3 297       /* re */
#define red3 313      /* re thang */
#define mi3 330       /* mi */
#define fa3 352       /* fa */
```

```

#define fad3 374 /* fa thang */
#define sol3 396 /* do thang */
#define sold3 415 /* sol */
#define la3 440 /* la */
#define lad3 468 /* la thang */
#define si3 495 /* si */
#define do4 528 /* do octave 4 */
#define dod4 565 /* do thang */
#define re4 594 /* re */
#define red4 625 /* re thang */
#define mi4 660 /* mi */
#define fa4 704 /* fa */
#define fad4 748 /* fa thang */
#define sol4 792 /* sol */
#define sold4 836 /* sol thang */
#define la4 880 /* la */
#define lad4 935 /* la thang */
#define si4 990 /* si */
#define do5 1050 /* do octave 5 */
#define lang 30000 /* dau lang la sieu am */

void not (unsigned ,float ); /* prototype not nhac */
int KhoiTaoDoHoa(void); /* prototype do hoa */

void main()
{
    /* Toa do hinh sao nam canh */
    int poly10[] = { 320,150,285,225,200,225,270,270,240,350,
                    320,300,390,350,360,270,480,225,345,225 };
    float d , /* not den */

```

```

tr, /* not trang */
tro, /* not tron */
md, /* moc don */
mk, /* moc kep */
m3, /* moc ba */
m4, /* moc bon */
dc, /* not den cham */
trc, /* not trang cham */
troc; /* not tron cham */
printf("\n Vao gia tri not den d = ");scanf("%f",&d);
clrscr();
KhoiTaoDoHoa();
setbkcolor(RED);
setcolor(YELLOW);
setfillstyle(SOLID_FILL,YELLOW);
fillpoly(10,poly10);
/* Truong do cac not nhac */
tr = 2*d; tro = 4*d; md = d/2;
mk = d/4; m3 = d/8; m4 = d/16;
dc = 3*d/2; trc = 3*d;troc = 6*d;

/* Bat dau ban nhac TIEN QUAN CA */

not(re2,d);not(mi2,d);not(re2,d);not(sol2,tr);not(sol2,tro);
not(la2,d);not(sol2,d);not(si2,tr);not(si2,tro);not(la2,d);
not(sol2,d);not(mi2,tr);not(sol2,tr);not(sol2,d);not(mi2,d);
not(re2,d);not(si1,d);not(re2,tro);not(sol2,d);not(la2,d);
not(si2,tr);not(si2,tr);not(si2,tr);not(la2,d);not(sol2,d);
not(re3,tro);not(si2,d);not(sol2,d);not(la2,tr);not(la2,tr);

```

```

not(si2,tr);not(fad2,d);not(re2,d);not(sol2,tro);not(si2,d);
not(do3,d);not(re3,tr);not(re3,tr);not(mi3,tro);not(re3,d);
not(si2,tro);not(si2,tr);not(la2,d);not(sol2,tr);not(re2,tr);
not(fad2,tr);not(fad2,d);not(la2,d);not(sol2,tro);not(si2,d);
not(do3,d);not(re3,tr);not(re3,tr);not(mi3,tro);not(re3,d);
not(si2,tro);not(si2,tr);not(la2,d);not(sol2,tr);not(sol2,tr);
not(re2,tro);not(re3,tro);not(si2,tr);not(sol2,tr);not(mi3,tro);
not(re3,tr);not(si2,d);not(la2,d);not(re2,d);not(la2,tr);
not(la2,tro);not(si2,tr);not(sol2,tro);
}

void not(unsigned tanso, float dai) /* Ham xac dinh not nhac */
{
sound(tanso);
delay(dai);
nosound();
}

int KhoiTaoDoHoa(void) /* Ham khoi tao do hoa */
{
int graphdriver = DETECT,graphmode,error_code;
initgraph(&graphdriver,&graphmode, "..\\bgi");
error_code = graphresult();
if (error_code != grOk)
{
return(-1);
if ((graphdriver != EGA ) && (graphdriver != VGA ))
{
closegraph();
}
}

```

```

        return(0);
    }
    return(1);
}

```

Bài số 101 Tạo cửa sổ ở chế độ văn bản

Bài giải Để khởi tạo chế độ văn bản ta sử dụng :

```

textmode ( mode );
int mode;

```

để chọn một trong 5 chế độ sau đây :

Ký hiệu	Giá trị số	Chế độ văn bản video
LASTMODE	-1	Chế độ văn bản có trước
BW40	0	Đen trắng 40 cột
C40	1	16 màu, 40 cột
BW80	2	Đen trắng 80 cột
C80	3	16 màu, 80 cột
MONO	7	Đơn sắc, 80 cột

Độc lập với chế độ văn bản được chọn màn hình luôn gồm 25 hàng

Đặt màu cửa sổ bằng `textbackground (color);`
`int color;`

Đặt màu chữ dùng `textcolor (color);`
`int color;`

Chương trình CT101 để kết thúc cuốn sách có cửa sổ gồm dòng chữ sau :

TRUONG DAI HOC BACH KHOA HA NOI

KHOA THIET BI DIEN

Cac tac gia chan thanh cam on ban doc

Neu ban co yeu cau chuong trinh xin moi

lien he voi KHOA THIET BI DIEN DT: 8692511

Chung toi xin dap ung mien phi

CAC TAC GIA

```
/* CT101 Chuong trinh lap cua so */
```

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
void main()
```

```
{
```

```
textmode(C40) ; /* che do van ban 16 mau 40 cot */
```

```
/* Man hinh mau tim */
```

```
textbackground(MAGENTA) ;
```

```
window(1,1,40,25) ; /* Man hinh 25 X 80 */
```

```
clrscr() ;
```

```
textcolor(WHITE) ; /* Dong chu mau trang */
```

```
printf("\n\n TRUONG DAI HOC BACH KHOA HA NOI ") ;
```

```
printf("\n\n\n KHOA THIET BI DIEN ") ;
```

```
printf("\n\n\n Cac tac gia chan thanh cam on ban doc ") ;
```

```
printf("\n\n Neu ban co yeu cau chuong trinh xin ") ;
```

```
printf("\n\n lien he KHOA THIET BI DIEN DT: 8692511 ") ;
```

```
printf("\n\n Chung toi xin dap ung mien phi ") ;
```

```
printf("\n\n CAC TAC GIA ") ;
```

```
getch();
```

```
return;
```

```
}
```

PHỤ LỤC

CÁC HÀM CỦA TURBO C

Hàm, thư mục	Cách sử dụng, ý nghĩa
<i>abs</i> math.h stdlib.h	int abs (int n) Trả về giá trị tuyệt đối của n
<i>acos</i> math.h	double acos (double x) Trả về $\arccos(x)$ nếu $-1 \leq x \leq 1$, nếu không 0
<i>asin</i> math.h	double asin (double x) Trả về $\arcsin(x)$ nếu $-1 \leq x \leq 1$, nếu không 0
<i>atan</i> math.h	double atan (double x) Trả về $\arctg(x)$
<i>atof</i> math.h stdlib.h	double atof (const char * s) Chuyển xâu ký tự s thành một số dấu phẩy động
<i>bioskey</i> bios.h math.h	int bioskey (int n) Quản lý bàn phím bằng BIOS n = 0 đọc một phím n = 1 thả một ký tự đã sẵn sàng n = 2 tình trạng các phím đã bị thay đổi
<i>alloc</i> alloc.h stdlib.h	void * alloc (size_t n, size_t t) Dành một khối nhớ nbx byte cho n đối tượng có kích thước t. Các byte của khối được khởi đầu bằng (\0)
<i>ceil</i> math.h	double ceil (double x) Làm tròn, trả về số nguyên nhỏ nhất lớn hơn hoặc bằng x
<i>cleardevice</i> graphics.h	void far cleardevice (void) Xóa màn hình đồ họa

<i>cloel</i> conio.h	void cloel (void) Xóa từ vị trí con trỏ đến cuối dòng trong cửa sổ hiện tại
<i>clrscr</i> conio.h	void clrscr (void) Xóa cửa sổ hiện tại và đặt con trỏ về góc trái trên (tọa độ 1,1)
<i>coreleft</i> alloc.h	unsigned coreleft (void) unsigned long coreleft (void) Trả về kích thước bộ nhớ RAM không sử dụng. Kết quả kiểu unsigned long với mô hình tiny, small, medium và kiểu unsigned long với các mô hình compact, large và huge
<i>cos</i> math.h	double cos (double x) Trả về cos (x)
<i>cosh</i> math.h	double cosh (double x) Trả về cosh (x)
<i>cprintf</i> conio.h	int cprintf (const char *f [, arg, ...]) In các đối số theo các đặc tả
<i>cputs</i> conio.h	int cputs (const char *s) In xâu ký tự s trong cửa sổ văn bản hiện tại
<i>exp</i> math.h	double exp (double x) Trả về luỹ thừa cơ số tự nhiên của x
<i>floor</i> math.h	double floor (double x) Trả về trị nguyên lớn nhất nhỏ hơn hoặc bằng x
<i>fabs</i> math.h	double fabs (double x) Trả về giá trị tuyệt đối của số thực x
<i>fillpoly</i> graphics.h	void far fillpoly (int n, int far *tp) Vẽ và tô một đa giác n đỉnh có các tọa độ sắp xếp trong tp
<i>free</i> alloc.h stdtib.h	void free (void *b) Giải phóng khỏi nhớ b

<i>ftime</i> sys\tmeb.h	void ftime (struct timeb *tb) Xác định ngày giờ hiện tại
<i>getmaxcolor</i> graphics.h	int far getmaxcolor (void) Lấy số màu cực đại trong chế độ đồ họa
<i>get maxx</i> graphics.h	int far getmaxx (void) Lấy hành độ cực đại của màn hình
<i>get maxy</i> graphics.h	int far getmaxy (void) Lấy tung độ cực đại của màn hình
<i>gettext</i> conio.h	int gettext (int ax, int ay, int bx, int by, void *t) Sao chép một hình chữ nhật của màn hình văn bản vào vùng nhớ đệm do t trả vào
<i>gettextinfo</i> conio.h	void gettextinfo (struct text_info *r) Lấy thông tin về chế độ video hiện thời
<i>gotoxy</i> conio.h	void gotoxy (int x, int y) Đặt con trỏ vào dòng y cột x trong cửa sổ
<i>graphresult</i> graphics.h	int far graphresult (void) Cho mã của lỗi đồ họa cuối cùng gặp phải
<i>initgraph</i> graphics.h	void far initgraph (&graphdriver,& graphmode,"") Khởi tạo hệ thống đồ họa, đặt card đồ họa, mode đồ họa với driver đã chọn
<i>int86</i> dos.h	int int86 (int n, union REGS *ar,union REGS *br) Ngắt logic số n
<i>isalnum</i> ctype.h	int isalnum (int c) Trả về một giá trị khác không nếu c là một chữ cái hoặc ký tự số. Trái lại sẽ trả về 0
<i>isdigit</i> ctype.h	int isdigit (int c) Trả về một giá trị khác không nếu c là một con số từ 0 đến 9
<i>kbhit</i> conio.h	int kbhit (void) Thử gõ một phím trên bàn phím, sẵn sàng trên màn hình

<i>line</i> graphics.h	void far line (int ax, int ay, int bx, int by) Vẽ một đoạn thẳng nối hai điểm (ax,ay) và (bx,by) Vị trí con trỏ không thay đổi
<i>linerel</i> graphics.h	void far linerel (int dx, int dy) Vẽ một đoạn thẳng từ vị trí hiện tại (x,y) đến điểm (x+dx,y+dy). Con trỏ được di chuyển tới vị trí mới
<i>lineto</i> graphics.h	void far lineto (int nx, int ny) Vẽ một đoạn thẳng từ vị trí hiện tại (x,y) đến điểm (nx,ny). Con trỏ được di chuyển tới điểm (nx,ny)
<i>log</i> math.h	double log (double x) Trả về $\ln(x)$, $x > 0$
<i>log10</i> math.h	double log10 (double x) Trả về $\log(x)$, $x > 0$
<i>memmove</i> mem.h	void *memmove (void *d, const void *s, size_t n) Sao chép một khối n byte của s vào trong d
<i>memset</i> mem.h	void *memset (void *s, int c, size_t n) Khởi đầu n byte của s với byte c
<i>moveto</i> graphics.h	void moveto (int nx, int ny) Chuyển vị trí hiện tại đến điểm có tọa độ (nx,ny)
<i>outtextxy</i> graphics.h	void outtextxy (int x, int y, char far *s) Ghi một xâu ký tự s vào điểm (x,y) trong cửa sổ đồ họa
<i>pow</i> math.h	double pow (double x, double y) Trả về x^y
<i>putch</i> conio.h	int putch (int c) Đưa một ký tự vào màn hình trong cửa sổ làm việc
<i>puttext</i> conio.h	int puttext (int ax, int ay, int bx, int by) Chép lại nội dung của vùng nhớ đệm t vào hình chữ nhật của màn hình văn bản
<i>random</i> stdlib.h	int random (int n) Phát một số nguyên ngẫu nhiên giữa 0 và n-1
<i>randomize</i> stdlib.h	void randomize (void) Khởi đầu bộ sinh số ngẫu nhiên

<i>realloc</i> stdlib.h alloc.h	void *realloc (void *b, size_t n) Điều chỉnh kích thước của khối b theo kích thước n
<i>rectangle</i> graphics.h	void far rectangle (int ax, int ay, int bx, int by) Vẽ một hình chữ nhật có góc trái trên (ax,ay) và góc phải dưới (bx,by)
<i>setbkcolor</i> graphics.h	void far setbkcolor (int c) Chọn màu nền cho màn hình đồ họa
<i>setcolor</i> graphics.h	void far setcolor (int c) Chọn màu vẽ
<i>setfillstyle</i> graphics.h	void far setfillstyle (int motif, int c) Đặt mẫu tô và màu tô đa giác
<i>sin</i> math.h	double sin (double x) Trả về sin(x)
<i>sinh</i> math.h	double sinh (int x) Trả về sh(x)
<i>sprintf</i> stdio.h	int sprintf (char *b, const char *f [, arg, ...]) Ghi các đối số trong xâu ký tự b theo đặc tả f
<i>sqrt</i> math.h	double sqrt (double x) Trả về căn bậc hai của x
<i>strcat</i> string.h	char *strcat (char *d, const char *s) Sao chép xâu nguồn s vào xâu nhận d
<i> strchr</i> string.h	char *strchr (const char *s, int c) Tìm lần xuất hiện của ký tự c trong xâu s
<i>strcmp</i> string.h	int strcmp (const char *sa, const char *sb) So sánh xâu sa và sb. Trả về 0 nếu các xâu giống nhau, trả về giá trị âm nếu sa nhỏ hơn sb và giá trị dương nếu sa lớn hơn sb
<i>strcpy</i> string.h	char *strcpy (char *d, const char *s) Sao chép xâu nguồn s vào cuối xâu nhận d
<i> strdup</i> string.h	char *strdup (const char *s) Dành vị trí và sao chép xâu tạo nên bản sao

<i>strlen</i> <i>string.h</i>	<code>size_t strlen (const char *s)</code> Trả về độ dài của xâu s, không tính đến ký tự không
<i>strncat</i> <i>string.h</i>	<code>char *strncat (char *d, const char *s, size_t n)</code> Mặc nối n byte của xâu nguồn s vào xâu nhận d
<i>strncpy</i> <i>string.h</i>	<code>char *strncpy (char *d, const char *s, size_t l)</code> Sao chép l byte của xâu nguồn s vào xâu nhận d bằng cách cắt d hay thêm các ký tự không
<i>strnset</i> <i>string.h</i>	<code>char *strnset (char *s, int c, size_t n)</code> Khởi đầu n ký tự đầu của xâu s với ký tự c
<i>strtod</i> <i>stdlib.h</i>	<code>double *strtod (const char *s, char **p)</code> Chuyển xâu s thành giá trị kiểu double, p trả về ký tự đầu tiên gây dừng
<i>tan</i> <i>math.h</i>	<code>double tan (double x)</code> Trả về tg(x)
<i>toupper</i> <i>ctype.h</i>	<code>int toupper (int c)</code> Chuyển chữ thường từ a đến z thành chữ hoa
<i>va_arg</i> <i>stdarg.h</i>	<code>type va_arg (va_list pvar, type)</code> Đổi số của danh sách thông số biến thiên pvar là kiểu type
<i>va_end</i> <i>stdarg.h</i>	<code>void va_end (va_list pvar)</code> Dành cho lỗi ra bình thường của một hàm đổi số thay đổi
<i>va_start</i> <i>stdarg.h</i>	<code>void va_start (va_list pvar, pfixe)</code> Đặt vị trí pvar vào đổi số thay đổi đầu tiên sau đổi số cố định cuối cùng pfixe
<i>wherex</i> <i>conio.h</i>	<code>int wherex (void)</code> Trả về cột của con trỏ trong cửa sổ làm việc
<i>wherey</i> <i>conio.h</i>	<code>int wherey (void)</code> Trả về dòng của con trỏ trong cửa sổ làm việc
<i>window</i> <i>conio.h</i>	<code>void window (int ax, int ay, int bx, int by)</code> Định cửa sổ văn bản ở góc trái trên tọa độ (ax,ay) đến góc phải dưới tọa độ (bx,by)

TÀI LIỆU THAM KHẢO

- [1] 101 thuật toán và chương trình bài toán khoa học và kinh tế
ngôn ngữ PASCAL, Lê Văn Doanh - Trần Khắc Tuấn, NXB KHTT
1993
- [2] Exercices en Langage C. Claude Delannoy, Eyrolles 1992
- [3] TURBO C, G. Leblance, Eyrolles 1990
- [4] Outils mathématique en C++, Alain Reverchon, Dunod 1993
- [5] TURBO C des outils. Albert Levine, Berti Editions 1993
- [6] TURBO C++ Bible. Nabaiyoti - Barkakati, The Waite
Group's UNIX 1991
- [7] The C proramming language, B. Kernighan and D. Ritchie,
Prentice Hall 1989
- [8] TURBO C. Richard Winner (Bản tiếng Nga) 1992
- [9] Programmer's guide Borland C++ Version 4.0, Borland
International, Inc 1993

LÊ VĂN DOANH TRẦN KHẮC TUẤN
LÊ ĐÌNH ANH

**101 THUẬT TOÁN VÀ CHƯƠNG TRÌNH
BÀI TOÁN KHOA HỌC KỸ THUẬT VÀ KINH TẾ
BẰNG NGÔN NGỮ C**

Chịu trách nhiệm xuất bản : Pgs.Pts. TÔ ĐĂNG HẢI
Biên tập : HỒNG VŨ
Sửa bản in : XUÂN PHƯƠNG
Vẽ bìa : TRẦN THÁNG

**NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
HÀ NỘI – 2006**

In 500 cuốn, khổ 14,5 x 20,5 cm, tại Xưởng in NXB Văn hoá dân tộc.
Giấy phép xuất bản số: 546 - 40 - 6 / 10 / 2005.
In xong và nộp lưu chiểu tháng 12 / 2005.

205339



8 935048 953393

Giá: 30.000đ