

BÀI GIẢNG

# CẤU TRÚC DỮ LIỆU

(BẬC CAO ĐẲNG)

Chương 1: NHẬP MÔN CẤU TRÚC DỮ LIỆU

Nguyễn Thanh Cẩm

# NỘI DUNG TRÌNH BÀY

---

1. Ý nghĩa cấu trúc dữ liệu
2. Cấu trúc dữ liệu và các vấn đề liên quan
3. Thuật toán

# 1. Ý nghĩa cấu trúc dữ liệu

---

**DATA STRUCTURE + ALGORITHM = PROGRAM**

*Niklaus wirth*

- Để giải bài toán trên máy tính: cần thuật toán
  - Thuật toán phản ánh phép xử lý
  - Dữ liệu biểu diễn thông tin cần thiết của bài toán
  - vd:
- 
- Cấu trúc dữ liệu thay đổi → thuật toán thay đổi theo
-

## 2. Cấu trúc dữ liệu và các vấn đề liên quan

---

- a. Dữ liệu và lưu trữ dữ liệu
  - a. Các kiểu dữ liệu đơn giản
  - b. Các kiểu dữ liệu cấu trúc

## 2. Cấu trúc dữ liệu và các vấn đề liên quan

---

### a. Dữ liệu và lưu trữ dữ liệu

Dữ liệu là vật mang thông tin đã được chuẩn hóa.

Cần phân biệt dữ liệu với thông tin:

- Dữ liệu tồn tại khác quan
  - Thông tin có ý nghĩa chủ quan.
-

## 2. Cấu trúc dữ liệu và các vấn đề liên quan

---

### a. Dữ liệu và lưu trữ dữ liệu

Trong một bài toán, dữ liệu gồm một tập các phần tử cơ sở, gọi là dữ liệu nguyên tử. Nó có thể là một chữ số, một ký tự, một từ,...tùy vào bài toán cụ thể

Trên cơ sở các dữ liệu nguyên tử, các cung cách liên kết chúng với nhau sẽ dẫn tới các cấu trúc dữ liệu khác nhau

---

## 2. Cấu trúc dữ liệu và các vấn đề liên quan

---

### a. Dữ liệu và lưu trữ dữ liệu

- Khi chọn một cấu trúc dữ liệu phải nghĩ ngay tới các phép toán tác động lên cấu trúc ấy và ngược lại
  - Cách biểu diễn một cấu trúc dữ liệu trong bộ nhớ được gọi là cấu trúc lưu trữ (storage structure)
  - Có thể có nhiều CTLT khác nhau cho cùng một CTDL, cũng có thể có nhiều CTDL khác nhau mà được cài đặt trong bộ nhớ bởi cùng một kiểu cấu trúc lưu trữ
  - CTDL trong và CTDL ngoài
-

## 2. Cấu trúc dữ liệu và các vấn đề liên quan

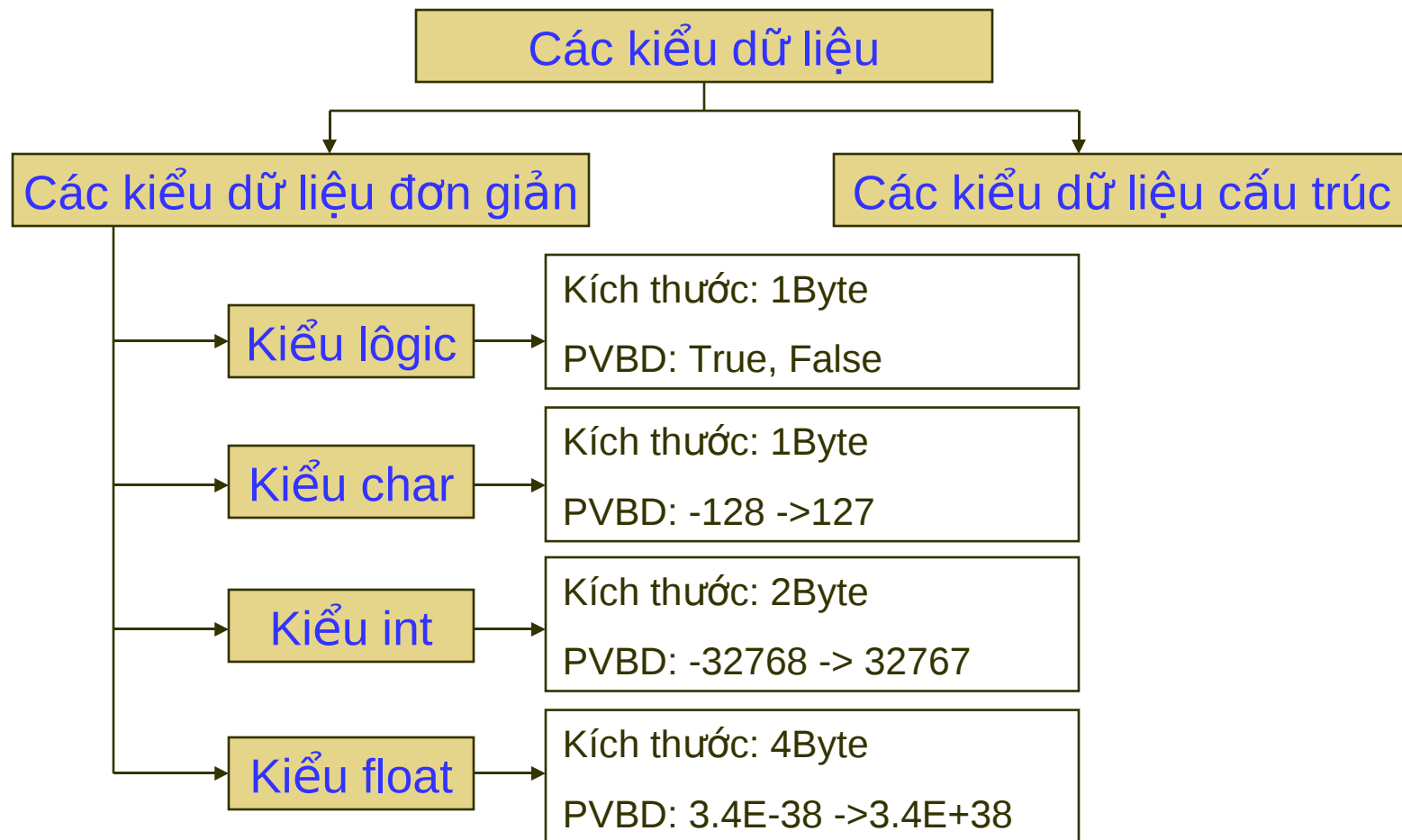
---

- a. Dữ liệu và lưu trữ dữ liệu
- a. Các kiểu dữ liệu đơn giản
- a. Các kiểu dữ liệu cấu trúc



## 2. Cấu trúc dữ liệu và các vấn đề liên quan

### b. Dữ liệu và lưu trữ dữ liệu



## 2. Cấu trúc dữ liệu và các vấn đề liên quan

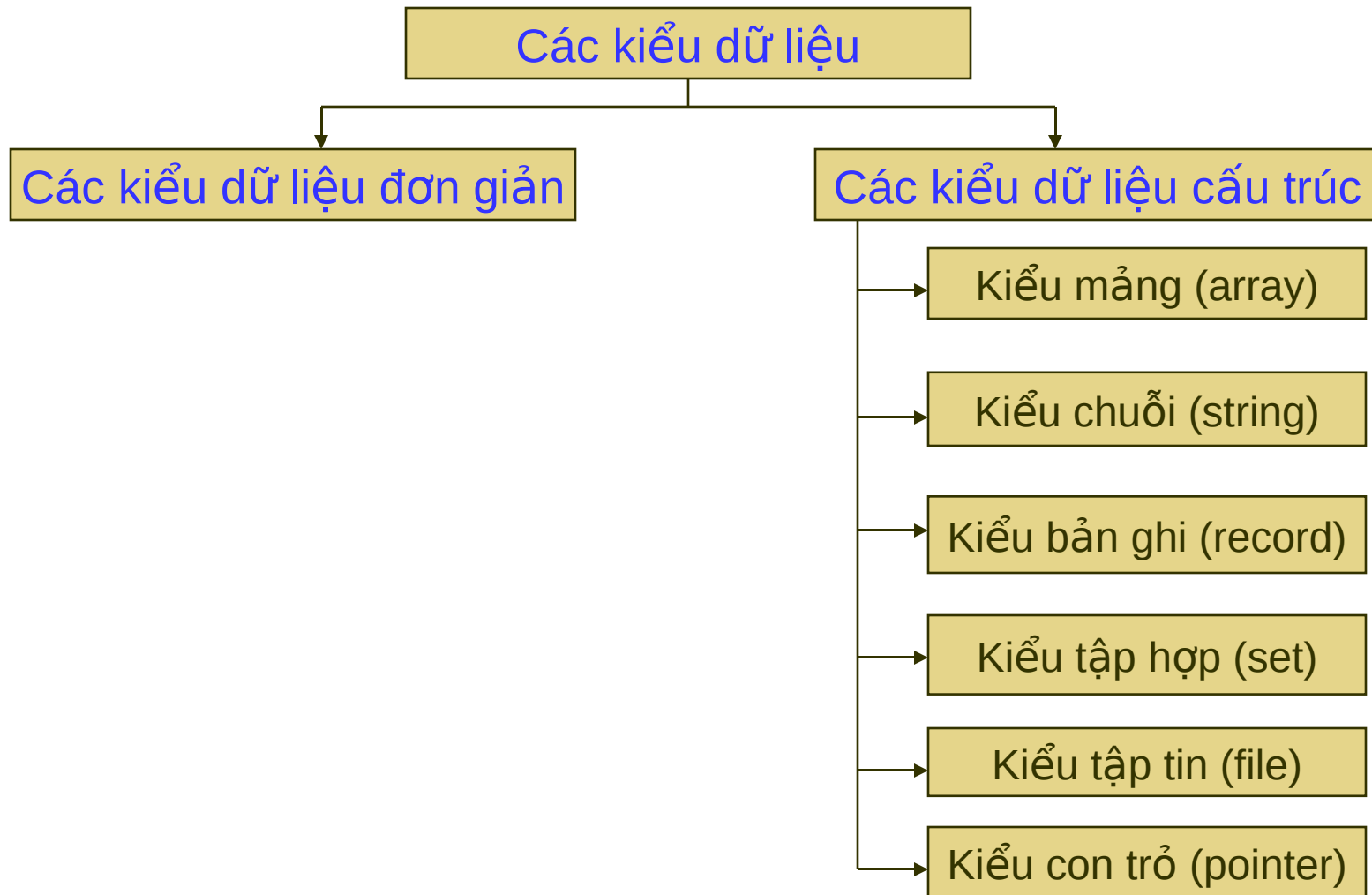
---

- a. Dữ liệu và lưu trữ dữ liệu
- b. Các kiểu dữ liệu đơn giản
  - a. Các kiểu dữ liệu cấu trúc

## 2. Cấu trúc dữ liệu và các vấn đề liên quan

---

### c. Các kiểu dữ liệu cấu trúc



## 3. Thuật toán

---

### a. Định nghĩa

a. Các cấu trúc điều khiển thuật toán

b. Chương trình con

---

## 3. Thuật toán

---

### a. Định nghĩa

Thuật toán là tập hợp hữu hạn các thao tác dẫn đến lời giải cho một vấn đề hay bài toán nào đó trong thời gian hữu hạn.

Các tính chất cơ bản của thuật toán:

- Tính đúng đắn
  - Tính hữu hạn
  - Tính tất định
  - Tính hiệu quả
  - Tính dễ hiểu
-

### 3. Thuật toán

---

#### **b. Các cấu trúc điều khiển thuật toán**

Cấu trúc tuần tự

Cấu trúc lặp

Cấu trúc chọn

---

## 3. Thuật toán

---

### b. Các cấu trúc điều khiển thuật toán

#### Cấu trúc tuần tự

- Lệnh gán
  - Lệnh hợp thành
  - Thủ tục
-

## 3. Thuật toán

---

### b. Các cấu trúc điều khiển thuật toán

#### Cấu trúc lặp

- Cấu trúc for
  - Cấu trúc while
  - Cấu trúc do while
-



## 3. Thuật toán

---

### b. Các cấu trúc điều khiển thuật toán

#### Cấu trúc chọn

- Cấu trúc if
- Cấu trúc switch

## 3. Thuật toán

---

### c. Chương trình con

Hàm có kiểu (hàm)

Hàm không kiểu (thủ tục)

---

## 3. Thuật toán

---

### c. Chương trình con

#### Hàm có kiểu

Là chương trình con, xử lý tính toán với mục đích trả về giá trị của đối tượng nào đó

#### Khai báo:

```
<Kiểu> <tên_hàm>(danh sách tham số hình thức)
{
...
return KQ;
}
```

#### Gọi hàm:

```
tên_hàm(danh sách tham số thực sự);
```

---

## 3. Thuật toán

---

### c. Chương trình con

#### Hàm không kiểu (thủ tục)

Là chương trình con, xử lý tính toán một công việc nào đó

#### Khai báo:

```
<Kiểu> <tên_hàm>(danh sách tham số hình thức)  
{  
...  
}
```

#### Gọi hàm:

```
Tên_hàm(danh sách tham số thực sự);
```

---

# KHOA KHOA HỌC MÁY TÍNH – BỘ MÔN LẬP TRÌNH

---

*Chúc các bạn thành công !*

---

# BÀI TẬP

---

1. Viết chương trình giải phương trình bậc nhất  $ax + b = 0$ .
  2. Viết chương trình tìm UCLN và BCNN của hai số nguyên dương nhập từ bàn phím.
  3. Viết chương trình tính tích hai ma trận  $A(n \times m)$  và  $B(m \times p)$ . In kết quả ma trận tích  $C(n \times p)$ .
  4. Viết một hàm thay thế một chuỗi con trong một chuỗi bằng một chuỗi con khác.
  5. Dùng cấu trúc mảng. Viết chương trình quản lý danh sách sinh viên (gồm họ tên, mã sinh viên). Chương trình có khả năng thêm, bớt, tìm kiếm.
-

## 1. Viết chương trình giải phương trình bậc nhất $ax + b = 0$

---

```
/* Bai tap1 - Giai phuong trinh bac nhat AX + B = 0 */
#include <stdio.h>
void main()
{
    float a, b;
    printf("\nGiai phuong trinh bac nhat AX + B = 0");
    printf("\nCho biet cac he so A B : ");
    scanf("%f%f", &a, &b);
    if (a==0)
        if (b!=0)
            printf("Phuong trinh vo nghiem");
        else
            printf("Phuong trinh co vo so nghiem");
    else
        printf("Nghiem x = %f", -b/a);
    getch();
}
```

## 2. Viết chương trình tìm UCLN và BCNN của hai số nguyên dương nhập từ bàn phím.

---

```
#include <stdio.h>
unsigned int UCLN (unsigned int n, unsigned int m)
{ while (n != 0 && m != 0)
  if (n>m)    n -= m;
  else       m -= n;
  if (n == 0) return m;
  else       return n;
}
Unsigned int BCNN (unsigned int n, unsigned int m)
{ return n * m / UCLN(n, m);
}
void main()
{ unsigned int n, m;
  printf("\nNhap hai vao so nguyen duong : ");
  scanf("%u%u", &n, &m);
  printf("\nUCLN cua %u va %u = %u", n, m, UCLN(n,m));
  printf("\nBCNN cua %u va %u = %u", n, m, BCNN(n,m));
  getch();
}
```

---



### 3. Viết chương trình tính tích hai ma trận $A(n \times m)$ và $B(m \times p)$ . In kết quả ma trận tích $C(n \times p)$ .

---

```
#include <stdio.h>
#define MAX 10
void in_ma_tran(int A[MAX][MAX], int n, int m, char id)
{ int i, j;
  printf("\nMa tran %c : ", id);
  for (i=0; i<n; i++)
  { printf("\n");
    for(j=0; j<m; j++)
      printf("%3d ",A[i][j]);
    }
}
void nhap_so_nguyen(int *value, int min, int max, char id)
{ do {
  printf("\nNhap gia tri %c (trong khoang %d den %d)", id, min,
max);
  scanf("%d", value);
}
while (*value < min || *value > max);
}
```

---

### 3. Viết chương trình tính tích hai ma trận $A(n \times m)$ và $B(m \times p)$ . In kết quả ma trận tích $C(n \times p)$ .

---

```
void main()
{ int A[MAX][MAX], B[MAX][MAX], C[MAX][MAX], n, m, p, i, j, k;
  nhap_so_nguyen(&n, 2, MAX, 'n'); nhap_so_nguyen(&m, 2, MAX, 'm');
  nhap_so_nguyen(&p, 2, MAX, 'p'); printf("\nNhap ma tran A : ");
  for (i=0; i<n; i++)
    for (j=0; j<m; j++)
      { printf("A[%d,%d] = ", i, j);   scanf("%d", &(A[i][j]));
        } printf("\nNhap ma tran B : ");
  for (i=0; i<m; i++)
    for (j=0; j<l; j++)
      {   printf("B[%d,%d] = ", i, j);   scanf("%d", &(B[i][j]));   }
  in_ma_tran(A, n, m, 'A'); in_ma_tran(B, n, m, 'B');
  for (i=0; i<n; i++)
    for (j=0; j<p; j++)
      { C[i][j] = 0;
        for (k=0; k<m; k++) C[i][j] += A[i][k] * B[k][j];
      } in_ma_tran(C, n, m, 'C');
  getch();
}
```

---

#### 4. Viết một hàm thay thế một chuỗi con trong một chuỗi bằng một chuỗi con khác.

---

```
/* Thay the chuoai con trong mot chuoai bang mot chuoai con khac */
#include <stdio.h>
#include <string.h>
#include <alloc.h>
char *tim_thay(char *source, char *substr, char *replace)
{ char *found, *temp, *stemp; int pos = 0;
  stemp = strdup(source); found = strstr(stemp + pos, substr);
  while (found)
  { pos = found - stemp + strlen(replace) - strlen(substr) + 1;
    temp = (char *) malloc(sizeof(stemp) + strlen(replace) -
strlen(substr) + 1);
    strncpy(temp, stemp, found - stemp);
    temp[found-stemp] = 0;  strcat(temp, replace);
    strcat(temp, found + strlen(substr));
    free(stemp);  stemp = (char *)malloc(sizeof(temp) + 1);
    strcpy(stemp, temp);  free(temp);
    found = strstr(stemp + pos, substr);
  } return stemp;
}
```

---

#### 4. Viết một hàm thay thế một chuỗi con trong một chuỗi bằng một chuỗi con khác.

---

```
void main()
{
    char source[255], substr[50], replace[50], *result;

    printf("\nNhap chuoai nguon : ");
    gets(source);
    printf("\nNhap chuoai tim kiem : ");
    gets(substr);
    printf("\nNhap chuoai thay the : ");
    gets(replace);
    result = tim_thay(source, substr, replace);
    printf("\nKet qua = %s", result);
    getch();
}
```

5. Đúng cấu trúc mảng. Viết chương trình quản lý danh sách sinh viên (gồm họ tên, mã sinh viên). Chương trình có khả năng thêm, bớt, tìm kiếm.

---

```
#include <stdio.h>
#include <ctype.h>
#include <mem.h>
#include <string.h>
```

```
#define MAX 100
#define TOAN 0
#define LY 1
#define HOA 2
```

```
struct sinhvien {
    char masv[5];
    char hoten[35];
    float diem[3];
} danhsach[MAX];
int n = 0;
```

---

5. Đúng cấu trúc mảng. Viết chương trình quản lý danh sách sinh viên (gồm họ tên, mã sinh viên). Chương trình có khả năng thêm, bớt, tìm kiếm.
- 

```
void nhapmoi()
{ char masv[5], tmp[3];
  int i; float diem[3];
  do {
    printf("\nCho biet ma sinh vien: ");
    gets(masv);
    if (strlen(masv))
    { strcpy(danh sach[n].masv, masv);
      printf("\nCho biet ho ten : ");
      gets(danh sach[n].hoten);
      printf("\nCho biet diem so : ");
      for (i=0; i<3; i++)
      { scanf("%f", &diem[i]);
        danh sach[n].diem[i] = diem[i];
      } gets(tmp);
      n++;
    }
  } while (strlen(masv));
}
```

---

5. Đúng cấu trúc mảng. Viết chương trình quản lý danh sách sinh viên (gồm họ tên, mã sinh viên). Chương trình có khả năng thêm, bớt, tìm kiếm.

---

```
void timkiem()
{ char masv[5]; int i = 0, found = 0;
  printf("\nCho biet ma so lop : ");
  gets(masv);
  if (strlen(masv))
    while (i<n)
      if (strcmp(danhsach[i].masv, masv) == 0)
        { printf("\nMa so lop : %s", danhsach[i].masv);
          printf("\nHo va ten : %s", danhsach[i].hoten);
          printf("\nDiem Toan : %f", danhsach[i].diem[TOAN]);
          printf("\nDiem Ly   : %f", danhsach[i].diem[LY]);
          printf("\nDiem Hoa  : %f", danhsach[i].diem[HOA]);
          found = 1;
          break;
        }
      else i++;
  if (!found)
    printf("\nKhong tim thay!!!");
}
```

---

5. Đúng cấu trúc mảng. Viết chương trình quản lý danh sách sinh viên (gồm họ tên, mã sinh viên). Chương trình có khả năng thêm, bớt, tìm kiếm.

---

```
void xoa()
{ char masv[5], traloi; int i = 0, j;
  printf("\nCho biet ma sinh vien : "); gets(masv);
  if (strlen(masv))
    while (i<n)
      if (strcmp(danhsach[i].masv, masv) == 0)
        { printf("\nMa so lop : %s", danhsach[i].masv);
          printf("\nHo va ten : %s", danhsach[i].hoten);
          printf("\nDiem Toan : %f", danhsach[i].diem[TOAN]);
          printf("\nDiem Ly   : %f", danhsach[i].diem[LY]);
          printf("\nDiem Hoa  : %f", danhsach[i].diem[HOA]);
          printf("\nCo muon xoa khong (C/K)? ");
          do { traloi = toupper(getch());
              } while (traloi != 'C' && traloi != 'K');   putc(traloi,
stdout);
          if (traloi == 'C')
            { n--; memcpy(&danhsach[i], &danhsach[i+1],
sizeof(struct sinhvien) * (n-i));
              break;
            } }
        else    i++;
    }
```

---



5. Đúng cấu trúc mảng. Viết chương trình quản lý danh sách sinh viên (gồm họ tên, mã sinh viên). Chương trình có khả năng thêm, bớt, tìm kiếm.

---

```
void menu()
{ printf("\n*****"); printf("\n* 1. Them  *");
  printf("\n* 2. Xoa  *"); printf("\n* 3. Tim kiem *");
  printf("\n* 0. Thoat  *"); printf("\n*****");
  printf("\nChon lua ? ");
}
void main()
{ char traloi;
  do { menu();
    do { traloi = getch();
      } while (traloi < '0' || traloi > '3');
    putc(traloi, stdout);
    switch (traloi)
    { case '1' : nhapmoi(); break;
      case '2' : xoa(); break;
      case '3' : timkiem(); break;
    }
  } while (traloi != '0');
}
```

---