3 Mảng – array

Nội dung

- Mảng một chiều
- Một số ví dụ
- Khởi tạo mảng
- Mảng ký tự
- Mảng nhiều chiều

3. Mảng – array

- **Bài toán**: Điểm môn THDC của các thành viên trong lớp được nhập vào từ bàn phím. Hãy sắp xếp và đưa ra các điểm theo thứ tự tăng dần.

```
printf ("Nhap diem thu 1\n");
scanf ("%f", &diem1);
printf ("Nhap diem thu 2\n");
scanf ("%f", &diem2);
. . .
```

3. Mảng – array

Mảng : là một tập hợp hữu hạn các phần tử có cùng kiểu dữ liệu được lưu trữ kế tiếp nhau trong bộ nhớ.

Khai báo mảng:

kiểu_dữ_liệu tên_biến_mảng[số_phần_tử];

- VD.

```
int  A[10];
float  bang_diem[50];
char bang_ky_tu[26];
```

3. Mảng – array

Truy cập vào một phần tử trong mảng :

- **`bang_diem[5]`** : phần tử có chỉ số 5 trong mảng **`bang_diem`**

tên_biến_mảng[chỉ_số]

- **Chú ý**:
  - Phần tử đầu tiên trong mảng có chỉ số là 0.
  - **`bang_diem[5]`** sẽ là phần tử thứ 6 trong mảng.
  - Phần tử cuối cùng trong mảng có chỉ số là **`số_phần_tử-1`**

3. Mảng – array

- Thao tác với các phần tử trong mảng như với số các biến thông thường khác.

```
bang_diem[3]=7;
```

```
printf("Nhap vao diem thu 5: ");
scanf("%f",&bang_diem[4]);
```

```
bang_diem[5] = bang_diem[3] +1;
```

```
printf("Diem thanh vien thu 7: %.2f",
bang_diem[6]);
```

3. Mảng – array

- Các phần tử trong mảng được lưu trữ liên tục trong bộ nhớ

```
int values[10];
```

value [0]	
value [1]	
value [2]	
value [3]	
value [4]	
value [5]	
value [6]	
value [7]	
value [8]	
value [9]	

3. Mảng – array

```
int values[10];

values[0] = 197;
values[2] = -100;
values[5] = 350;
values[3] =
  values[0] +
  values[5];
values[9] =
  values[5] / 10;
--values[2];
```

value [0]	197
value [1]	
value [2]	-101
value [3]	547
value [4]	
value [5]	350
value [6]	
value [7]	
value [8]	
value [9]	35

```
#include <stdio.h>

int main (void)
{
    int values[10];
    int index;
    values[0] = 197;
    values[2] = -100;
    values[5] = 350;
    values[3] = values[0] + values[5];
    values[9] = values[5] / 10;
    --values[2];

    for ( index = 0; index < 10; ++index )
       printf ("values[%i] = %i\n", index,
  values[index]);

    return 0;
}
```

3. Mảng – array

Sử dụng mảng như bộ đếm:

- VD. Để khảo sát chất lượng một loại sản phẩm mới nhà sản xuất đưa ra tiêu chí đánh giá chất lượng sản phẩm theo giá trị từ 0 đến 5 (0 là không biết, 1 là rất tồi, 2 là tồi, 3 là trung bình, 4 là tốt và 5 là rất tốt).

  Các tiêu chí này được khách hàng đánh giá thông qua một cuộc khảo sát tại một siêu thị, khoảng 5000 người đã được phỏng vấn.

  Bây giờ ta muốn thống kê kết quả của cuộc khảo sát.

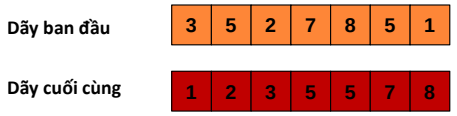
```
#include <stdio.h>
int main (void)
{
    int ratingCounters[6], i, response;

    for ( i = 0; i <= 5; ++i ) ratingCounters[i] = 0;

    printf ("Tra loi cua ban\n");
    for ( i = 1; i <= 20; ++i ) {
        scanf ("%i", &response);
        if ( response < 0 || response > 5 )
           printf ("Tra loi sai: %i\n", response);
        else
           ++ratingCounters[response];
    }
    printf ("\n\nLoai       So luong\n");
    printf ("------ -------------------\n");
    for ( i = 0; i <= 5; ++i )
        printf ("%4i%14i\n", i, ratingCounters[i]);
  return 0;
}
```

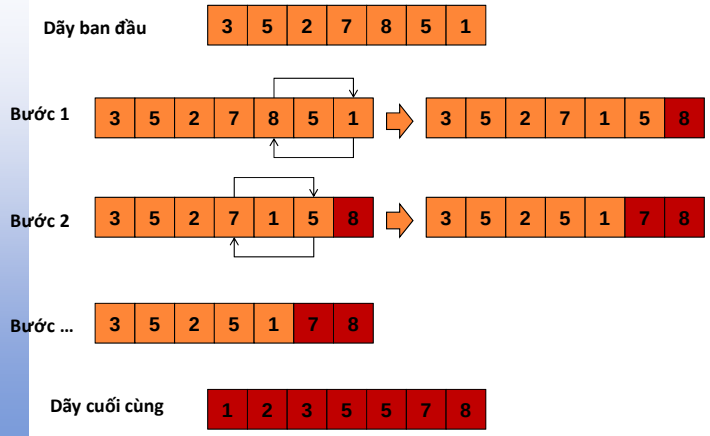
3. Mảng – array

- **Sắp xếp mảng**: sắp xếp các phần tử của mảng theo thứ tự tăng dần (hoặc giảm dần).



- **Các thuật toán sắp xếp**: sắp xếp chèn, lựa chọn, nổi bọt, shellsort, quicksort, mergesort, heapsort,….

Thuật toán sắp xếp lựa chọn



Thuật toán sắp xếp lựa chọn

- **Thuật toán sắp xếp lựa chọn**:
  - Nếu mảng ban đầu có từ 2 phần tử trở lên (n>=2)
    - **Gán giá trị k**: k=n
    - **Lặp**: cho đến khi k=1 thì dừng
      - Tìm phần tử có giá trị lớn nhất trong k phần tử ban đầu.
      - Đổi chỗ phần tử lớn nhất với phần tử thứ k
      - **Giảm k**: k=k-1

Thuật toán sắp xếp lựa chọn

```
#include <stdio.h>
int main (void)
{
    int A[10]={1,4,2,8,12,4,28,4,23,10};
    int i,k=10,tmp;
    int viTriMax;

    //in ra gia tri mang ban dau
    for(i=0;i<10;i++)
      printf("%d ",A[i]);

     printf("\n");
```

```
  while(k>1)
    {
        //tim gia tri lon nhat trong k phan tu
        viTriMax=0;
        for(i=1;i<k;i++)
            if(A[i]>A[viTriMax]) viTriMax=i;

        //doi cho voi phan tu thu k
        tmp=A[viTriMax];
        A[viTriMax]=A[k-1];
        A[k-1]=tmp;

        //giam  k
        k=k-1;
    }
  for(i=0;i<10;i++) printf("%d ",A[i]);
  printf("\n");
  return 0;
}
```

Tìm kiếm trên mảng

- **Bài toán**: Tìm kiếm trên mảng
  - **Đầu vào**: Cho một mảng gồm n phần tử, và một giá trị khóa k nào đó.
  - **Đầu ra**: Trả lời câu hỏi k có xuất hiện trong mảng
- Có thể đầu ra sẽ là vị trí khóa k trong mảng hoặc số lần xuất hiện của k trong mảng.

Mảng ban đầu	3	5	2	7	8	5	1
Khóa k	5						

Câu trả lời: k có xuất hiện trong mảng

Tìm kiếm trên mảng

- **Thuật toán tìm kiếm tuần tự**: so sánh lần lượt từng phần tử trên mảng.
  - Nếu mảng có >0 phần tử
    - Gán giá trị biến found = 0 (để xác định xem đã tìm thấy hay chưa)
    - Lặp: trong khi found =0 và chưa xét đến phần tử cuối cùng
      - So sánh giá trị phần tử hiện tại với k.
        - » **Nếu** đúng bằng thì gán biến found = 1.
        - » **Ngược lại** thì chuyển sang phần tử kế tiếp.

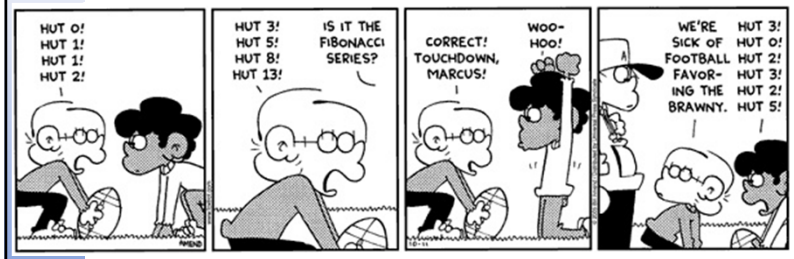
Tìm kiếm trên mảng

- Tìm kiếm quần tự:

```
int found =0;
i=0;
while(i<n && !found)
{
    if(A[i]==k) found = 1;
    else i++;
}
```

3. Mảng – array

- VD. **Sinh dãy số Fibonacci**



3. Mảng – array

```
// chuong trinh sinh 15 so Fibonacci dau tien
#include <stdio.h>
int main (void)
{
    int Fibonacci[15], i;
    Fibonacci[0] = 0; // theo dinh nghia
    Fibonacci[1] = 1;
    for ( i = 2; i < 15; ++i )
      Fibonacci[i] = Fibonacci[i-2] + Fibonacci[i-1];
    for ( i = 0; i < 15; ++i )
      printf ("%i\n", Fibonacci[i]);

    return 0;
}
```

3. Mảng – array

- VD. Sử dụng mảng để tạo ra các số nguyên tố. Số nguyên tố là số chỉ chia hết cho 1 và chính nó.
- Để kiểm tra một số n là nguyên tố:
  - **Thử chia cho các số từ 2 đến n-1**: **chỉ thực hiện được với n nhỏ, nếu n cỡ 10.000.000 thì quá chậm !**
  - Nếu n là nguyên tố thì nó cũng không chia hết cho các số nguyên tố khác.

➔ Hãy cài đặt để in sinh ra 100 số nguyên tố đầu tiên!

3. Mảng – array

Khởi tạo mảng

- **int integers[5] = { 0, 1, 2, 3, 4 };**
- **int integers[] = { 0, 1, 2, 3, 4 };**
- **char letters[5] = { 'a', 'b', 'c', 'd', 'e' };**
- **float sample[5] = {3*c, 3+x, 5.0, 4.6, 8.2};**
- **float sample_data[500] = { [2] = 500.5, [1] = 300.0, [0] = 100.0 };** **(*)**
- **int a[10] = { [9] = x + 1, [2] = 3, [1] = 2, [0] = 1 };** **(*)**

(*) chỉ có trong C99 (devC++,…)

3. Mảng – array

- **Mảng ký tự:**

```
#include <stdio.h>
int main (void)
{

    char word[] = { 'H', 'e', 'l', 'l', 'o', '!' };
    int i;
    for ( i = 0; i < 6; ++i )
       printf ("%c", word[i]);

    printf ("\n");

    return 0;
}
```

3. Mảng – array

- VD. Chương trình chuyển đổi cơ số, đổi số từ hệ cơ số 10 sang các hệ cơ số khác.

```
#include <stdio.h>
int main (void)
{
    const char baseDigits[16] = {
     '0', '1', '2', '3', '4', '5', '6', '7',
     '8', '9', 'A', 'B', 'C', 'D', 'E', 'F' };
    int convertedNumber[64];//so chuyen doi
    long int numberToConvert;//so ban dau
    int nextDigit, base, index = 0;

    // doc vao so va co so chuyen doi
    printf ("So can chuyen? ");
    scanf ("%ld", &numberToConvert);
    printf ("He co so moi? ");
    scanf ("%i", &base);
```

```
   // chuyen sang he co so moi
    do {
        convertedNumber[index] = numberToConvert %
 base;
        ++index;
        numberToConvert = numberToConvert / base;
    }
    while ( numberToConvert != 0 );

    // Hien thi ket qua the thu tu nguoc
    printf ("So da chuyen doi = ");
    for (--index; index >= 0; --index ) {
        nextDigit = convertedNumber[index];
        printf ("%c", baseDigits[nextDigit]);
    }

    printf ("\n");
    return 0;
}
```

3. Mảng – array

Mảng nhiều chiều: C cho phép khai báo mảng nhiều chiều.

- VD. Mảng hai chiều

Cột (j) / *Hàng (i)*	0	1	2	3	4
0	1	23	10	-3	0
1	3	1	78	9	2
2	0	23	0	34	1
3	100	9	123	39	-5

- Khai báo : `kiểu_phần_tử tên_mảng[số_hàng][số_cột];`
- Truy cập vào phần tử: `tên_mảng[i,j]`

3. Mảng – array

- **Khởi tạo mảng**
-
```
int M[4][5] = {
 { 10, 5, -3, 17, 82 },
 { 9, 0, 0, 8, -7 },
 { 32, 20, 1, 0, 14 },
 { 0, 0, 8, 7, 6 }
 };
```

- `int M[4][5] = { 10, 5, -3, 17, 82, 9, 0, 0, 8, -7, 32, 20, 1, 0, 14, 0, 0, 8, 7, 6 };`

3. Mảng – array

- VD. Thông tin về bán hàng của một của hàng tại các thời điểm trong ngày (sáng, chiều và tối) của một tuần được lưu vào một mảng.
  Hãy in ra thông tin về số lượng sản phẩm bán ra trung bình tại các thời điểm trong ngày của tuần đó

```
#include <stdio.h>
int main (void)
{
    int slBan[3][7];
    int i,j,tSang=0,tChieu=0,tToi=0;
    printf("Nhap so luong san pham ban trong tuan\n");
    for(j=0;j<7;j++)
    {
        printf("Ngay thu %i:",j+1);
        printf("So luong(Sang,chieu va toi ):");
        scanf("%i%i%i",&slBan[0][j],&slBan[1][j],
             &slBan[2][j]);
    }
    printf("\n\nThong ke trong tuan\n");
    for(j=0;j<7;j++) printf("Ngay %i  ",j+1);

    printf("\n");
    printf("-----------------------------------------
     ------------\n");
```

```
    for(j=0;j<7;j++) printf("%5i   ", slBan[0][j]);
    printf("\n");
    for(j=0;j<7;j++) printf("%5i   ",slBan[1][j]);
    printf("\n");
    for(j=0;j<7;j++) printf("%5i   ",slBan[2][j]);
    printf("\n");

    for(j=0;j<7;j++){
        tSang=tSang+slBan[0][j];
        tChieu=tChieu+slBan[1][j];
        tToi=tToi+slBan[2][j];
    }
    printf("\n\nTrung binh trong tuan:\n");
    printf("Sang: %.2f\n",(float)tSang/7);
    printf("Chieu: %.2f\n",(float)tChieu/7);
    printf("Toi: %.2f\n",(float)tToi/7);
    return 0;
}
```