



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

# C PROGRAMMING INTRODUCTION

## TUẦN 10: MẢNG

# Mảng

- Một khối gồm nhiều biến có cùng kiểu
- Mảng có thể được khai báo bằng kiểu bất kỳ
  - VD: `int Arr[10]` là mảng gồm 10 phần tử số nguyên
- VD:
  - Danh sách điểm của sinh viên
  - Chuỗi các số nhập vào bởi người dùng
  - Véc-tơ
  - Ma trận

# Mảng trong bộ nhớ

- Một chuỗi các biến có kiểu dữ liệu xác định
- Biến mảng được lưu trữ ở địa chỉ đầu tiên trong bộ nhớ
- VD:

```
int arr[10];
```



arr

- Phần tử thứ n của mảng `arr` được thể hiện bởi `arr[n-1]` **(0-based)**

# Khởi tạo

- Mảng có thể được khởi tạo khi khai báo
- Số lượng phần tử được khởi tạo không vượt quá kích thước mảng
  - các phần tử còn lại được khởi tạo bằng 0

# Khởi tạo (tiếp)

- Kích thước mảng có thể được xác định dựa trên khởi tạo
  - VD:
    - `int array1 [8] = {2, 4, 6, 8, 10, 12, 14, 16};`
    - `int array2 [] = {2, 4, 6, 8, 10, 12, 14, 16};`

# VD: Khởi tạo với vòng for

- Tạo một mảng số chẵn từ 2 đến 20. In ra nội dung của mảng

```
#include <stdio.h>
#define arraySize 10

int main()
{
    int s[ arraySize ];    // array S has 10 elements
    int i;
    for ( i = 0; i < arraySize; i++ )
        s[ i ] = 2 + 2 * i;
    printf("Element \t Value\n");
    for ( i = 0; i < arraySize; i++ )
        printf("%d\t%d\n", i, s[i]);

    return 0;
}
```

# Vào/ra DL với mảng

- Sử dụng vòng lặp for, tại mỗi bước lặp:
  - sử dụng hàm nhập DL như scanf
- hoặc
  - sử dụng hàm xuất DL như printf

month	rainfall (in mm)
1	40
2	45
3	95
4	130
5	220
6	210
7	185
8	135
9	80
10	40
11	45
12	30

*table of rainfall*

# VD

```
#include <stdio.h>
#define MONTHS 12

/* store and display rainfall in all months of the year */

int main()
{
    int rainfall[MONTHS];
    int i;

    for ( i=0; i < MONTHS; i++ ){
        printf("Enter the rainfall(mm):"); scanf("%d", &rainfall[i] );
    }
    /* Print from January to December */
    for ( i=0; i < MONTHS; i++ ) {
        printf( "%5d ", rainfall[i]);
    }
    printf("\n");
    return 0;
}
```



# Bài tập 10.1

- 1)Viết chương trình nhập mảng chứa 20 số nguyên
  - a) Tính tổng các số lẻ trong mảng
  - b) Tìm min

# Lời giải

```
#include <stdio.h>
#define MAX 20

int main()
{
    int a[MAX];
    int i,s,min;
    for ( i = 0; i < MAX; i++ ){
printf("a[%d]=",i);scanf("%d", &a[i]);
    }
    s=0; min=a[0];
    for ( i = 0; i < MAX; i++ ){
if (a[i] < min) min = a[i];
if (a[i]%2==1) s= s+ a[i];
    }
    printf("\nMinimum value in the array: %d",min);
    printf("Sum of all odd numbers: %d",s);
    return 0;
}
```

# Bài tập 10.2

- Cho một mảng gồm các phần tử được nhập vào bởi người dùng. Tính tổng của các cực trị địa phương trong mảng (cực trị địa phương là phần tử có giá trị lớn hơn phần tử liền trước và liền sau nó)

# Lời giải

```
.. // Array data input
sum=0;
for (i=1; i<=size-1; i++)
    if (a[i]>=a[i-1] && a[i]>=a[i+1])
        sum +=a[i];
```

# Mảng là tham số của hàm

- Hàm nhận tham số là mảng
- Thường cần thêm tham số là kích thước mảng (tại sao?)

# Mảng là tham số của hàm (tiếp)

- VD:

```
int calc_sum(int arr[], int size);
```

- Trong hàm **arr** được truy cập như bình thường
- Thay đổi trong hàm tạo ra thay đổi trong mảng (!)

# VD

```
int calc_sum(const int arr[], int size)
{
    int i = 0;
    int sum = 0;

    for (i = 0; i < size; ++i)
        sum += arr[i];

    return sum;
}
```

# Bài tập 10.3

- Cài đặt một hàm nhận tham số là hai mảng, trả về 1 nếu hai mảng giống nhau, 0 nếu ngược lại
- Viết chương trình sử dụng hàm này



# Lời giải

```
int compare_arrays(const int arr1[], const int
    arr2[], int size)
{
    int i = 0;

    for (i = 0; i < size; ++i)
    {
        if (arr1[i] != arr2[i])
            return 0;
    }

    /* if we got here, both arrays are identical */
    return 1;
}
```

# Bài tập 10.4

- Viết hai hàm:
  - sắp xếp các số nguyên trong mảng theo thứ tự giảm dần
  - sắp xếp các số lẻ trong mảng theo thứ tự giảm dần
- Viết chương trình yêu cầu người dùng nhập vào mảng gồm 10 phần tử và thực hiện hai hàm trên

# Lời giải

```
void DesSort (int a[], int n)
{
    int tmp;
    for (i = 0; i < n-1 ; i++)
        for (j = i+1; j < n; j++)
            if (a[i] < a [j]) {
                tmp=a[i];
                a[i]=a[j];
                a[j]= tmp;
            }
}
```

# Lời giải (tiếp)

```
void OddSort (int a[], int n)
{
    int tmp;
    for (i = 0; i < n-1 ; i++)
        for (j = i+1; j < n; j++)
            if (a[i]<a[j] && (a[i]%2) && (a[j]%2))
            {
                tmp=a[i];
                a[i]=a[j];
                a[j]= tmp;
            }
    }
}
```

# Bài tập 10.5

- Cho một mảng số nguyên:
  - a) Đếm số phần tử có giá trị bằng 0
  - b) Tìm độ dài của chuỗi dài nhất gồm các phần tử có giá trị bằng 0
  - c) Đếm số lần xuất hiện của các phần tử

# Lời giải

```
count=0;  
for (i=1; i<n; i++)  
    if (a[i] ==0)  
        count++;
```

# Lời giải (tiếp)

```
int max=0,temp=0;
for (i=0;i<n;i++)
{
    if (a[i]==0)
        temp=temp+1;
        if (a[i]!=0) {
            if (temp>max) {
                max=temp;
                temp=0;
            }
        }
}
if (temp>max) max=temp;
printf("Do dai day con bang 0 lon nhat
la:", max);
```

# Lời giải (tiếp)

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

```
void swap(int a[],int i, int j)
{
    int temp;
    temp = a[i];
    a[i] = a[j];
    a[j] =temp;
}
```



# Lời giải (tiếp)

```
void main()
{
    int n,a[100],i,j,dem;
    printf("Enter the number of elements in array:");
    scanf("%d",&n);
    for(i=0;i<n;i++) {
        printf("a[%d] = ",i); scanf("%d",&a[i]);
    }
    for(i=0;i<n-1;i++)
        for(j=i+1;j<n;j++)
            if (a[i]>a[j]) swap(a, i, j);

    i=0; j=0; dem=0;
    while (i<n) {
        dem=0;
        j=i;
        while (a[i]==a[j]) {
            dem++; j++;
        }
        printf("\n%d occurs for %d times in array",a[i],dem);
        i=j;
    }
}
```

# Mảng hai chiều

- Mảng của mảng:

```
int A[2][3] = { {1, 2, 3},  
               {4, 5, 6} };
```

1	2	3
4	5	6

- Mảng gồm 2 mảng số nguyên, mỗi mảng có 3 phần tử
- Truy cập: phần tử j của mảng i

**A[i][j]**

# VD : Cộng ma trận

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

```
#define SIZE 3
```

```
int main()
```

```
{
```

```
    int A[][SIZE] = {{1,2,3}, {4,5,6}, {7,8,9}};
```

```
    int B[][SIZE] = {{1,1,1}, {2,2,2}, {3,3,3}};
```

```
    int C[SIZE][SIZE];
```

```
    int i = 0, j = 0;
```

```
        for (i = 0; i < SIZE; ++i)
```

```
            for (j = 0; j < SIZE; ++j)
```

```
                C[i][j] = A[i][j] + B[i][j];
```

```
    return 0;
```

```
}
```

# Bài tập 10.6

- Viết chương trình khai báo 3 ma trận A, B, C kích thước 3x3 chứa số nguyên; khởi tạo A và B
- Tính kết quả của phép nhân ma trận và lưu vào C:  $C = A * B$
- In kết quả ra màn hình

# Lời giải

```
#include <stdio.h>

void main() {
    float a[3][3], b[3][3], c[3][3];
    int m,n,p;
    int i,j,k;
    float temp;
    for(i=0; i<3; i++){
        for (j=0; j<3; j++) {
            printf("a[%d][%d]=", i+1, j+1);
            scanf("%f", &temp);
            printf("b[%d][%d]=", i+1, j+1);
            scanf("%f", &temp);
            b[i][j]= temp;
        }
    }
}
```

# Lời giải (tiếp)

```
for(i=0; i<3; i++) {  
    for (j=0; j<3; j++) {  
        c[i][j]=0;  
        for (k=0; k<n; k++)  
            c[i][j]= c[i][j]+ a[i][k]*b[k][j];  
    }  
}  
printf("\n Matrix A:");  
for(i=0; i<3; i++) {  
    printf("\n");  
    for (j=0; j<3; j++) {  
        printf("%2.2f\t", a[i][j]);  
    }  
}
```

# Lời giải (tiếp)

```
printf("\n Matrix b:");  
for(i=0; i<3; i++) {  
    printf("\n");  
    for (j=0; j<3; j++) {  
        printf("%2.2f\t", b[i][j]);  
  
    }  
}  
printf("\n Matrix c:");  
for(i=0; i<3; i++) {  
    printf("\n");  
    for (j=0; j<3; j++) {  
        printf("%2.2f\t", c[i][j]);  
    }  
}
```

# Bài tập 10.7

- Nhập mảng  $n$  phần tử từ người dùng. Kiểm tra tính đối xứng của mảng



# Lời giải

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

```
int checkSymmetric(int a[], int n);
```

```
void main()
```

```
{
    int a[100],n,i;
    printf(" Number of elements: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("a[%d]= ",i);
        scanf("%d",&a[i]);
    }
    printf("\n Array's content:\n");
    for(i=0;i<n;i++)
        printf("%d ",a[i]);
    if(checkSymmetric(a,n))
        printf("\n array is symmetric");
    else
        printf("\n array is not symmetric ");
}
```

```
int checkSymmetric (int a[],int n)
{
    int i=0,j=n-1;
    while(i<=j)
    {
        if(a[i]!=a[j]) return 0;
        i++;
        j--;
    }
    return 1;
}
```

# Bài tập 10.8

- Viết hàm đảo ngược mảng. Sử dụng hàm này trong chương trình với mảng số thực được người dùng nhập vào.

# Lời giải

```
void reverse(float a[], int size)
{
    int i;
    float tmp;
    for (i=0; i<n/2; i++)
    {
        tmp=a[i]; a[i]=a[n-i-1];
        a[n-i-1]=tmp;
    }
}
```



25 YEARS ANNIVERSARY  
**SOICT**

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

**Thank you  
for your  
attentions!**

