

## Bài tập chương 1

**Bài 1.** Cho một mảng a có n số nguyên, hãy kiểm tra các điều kiện sau:

+ Mảng có sắp thứ tự tăng hay không?

(Áp dụng thuật toán đếm với điều kiện đếm là  $a[i] \leq a[i+1]$ )

+ Mảng có sắp thứ tự giảm hay không?

+ Mảng có sắp thứ tự hay không?

(Dựa vào kết quả của hai câu hỏi trên)

**Bài 2.** Đếm các số không âm trong mảng a có n số nguyên

```
int dem_khong_am ( int  *a, int  n)
{
    int i,kq=0;
    for (i = 0 ; i < n ; i++)    if (a[i] >= 0)  kq++;
    return  kq;
}
```

**Bài 3.** Đếm các số nguyên tố trong mảng a có n số nguyên

```
int dem_nguyen_to ( int  *a, int  n)
{
    int i,kq=0;
    for (i = 0 ; i < n ; i++)    if (nguyen_to(a[i]))  kq++;
    retrurn  kq;
}
```

**Bài 4.** Đếm số lần xuất hiện của phần tử x trong mảng a.

```
int count (int  x, int  *a, int  n)
{
    int i,kq=0;
    for (i = 0 ; i < n ; i++)
        if (a[i]==x)  kq++;
    return  kq;
}
```

**Bài 5.** Tìm số âm lớn nhất trong mảng a.

```
int tim_so_am (int  *a, int  n, int  &x)
{
    int i,kq = 0;
    for (i = 0 ; i < n  &&  kq== 0 ; i++)
        if (a[i] < 0) {  x = a[i];    kq = i;  }
    return  kq;
}
```

```

void max_am (int *a, int n)
{
    int i, x, max;
    if (tim_so_am(a,n,x)
    {
        max=x;
        for (i=0 ; i < n ; i++)
            if (max < a[i] && a[i] < 0) max = a[i];
        printf("\nSo can tim la %d : ",max);
    }
    else printf("\nMang nay khong co so am.");
}

```

**Bài 6.** Tìm phần tử có số lần xuất hiện nhiều nhất

```

int max_xuat_hien (int *a, int n)
{
    int i, max = a[0];
    for (i = 0 ; i < n ; i++)
        if (count(max, a, n) < count(a[i], a, n)) max = a[i];
    return max;
}

```

**Bài 7.** Tạo mảng b chứa tất cả các phần tử của mảng a sao cho mỗi phần tử chỉ xuất hiện trong b đúng một lần.

```

void mang_pt_xuat_hien_mot_lan (int *a, int n, int *b, int &m)
{
    int i;
    m = 0;
    for (i = 0 ; i < n ; i++)
        if (count(a[i], b, m) == 0) b[m++] = a[i];
}

```

**Bài 8.** Sắp thứ tự tăng các số dương và giữ cố định các số còn lại

```

void sort_duong (int *a, int n)
{
    int i, j;
    for (i = 0 ; i < n - 1 ; i++)
        for (j = i + 1 ; j < n ; j++)
            if (a[i] > a[j] && a[i] > 0 && a[j] > 0)
                swap(a[i], a[j]);
}

```

**Bài 9.** Sắp xếp mảng a sao cho: Các số chẵn đứng ở đầu mảng và có thứ tự tăng, các số lẻ đứng ở cuối mảng và có thứ tự giảm

**Bài 10.** Sắp xếp mảng a sao cho: Các số dương ở đầu mảng và có thứ tự tăng, Các số âm ở giữa mảng và có thứ tự giảm và các số 0 ở cuối mảng.

**Bài 11.** Viết hàm sắp xếp ma trận tăng dần từ trên xuống dưới và từ trái sang phải không dùng mảng phụ.

```
void SapTang(MATRAN a, int d, int c)
{
    for (int i = 0; i <= d*c-2; i++)
        for (int j = 0; j <= d*c-1; j++)
            if (a[i/c][i%c] < a[j/c][j%c])
            {
                int tmp = a[i/c][i%c] ;
                a[i/c][i%c] = a[j/c][j%c] ;
                a[j/c][j%c] = tmp ;
            }
}
```

**Bài 12.** Viết hàm tính tổng các phần tử trên cùng một dòng (cột) của ma trận.

**Bài 13.** Viết hàm tính tổng các giá trị lớn nhất trên mỗi dòng (cột) của ma trận.

**Bài 14.** Viết hàm tìm dòng (cột) có tổng lớn nhất (nhỏ nhất) trong ma trận.

**Bài 15.** Viết hàm sắp xếp ma trận theo thứ tự tăng dần từ trên xuống dưới và từ trái qua phải.

HD: Đảo ma trận sang mảng một chiều, sắp xếp trên mảng một chiều theo thứ tự tăng dần, sau đó chuyển ngược mảng một chiều thành ma trận kết quả.

**Bài 16.** Viết hàm sắp xếp ma trận theo thứ tự giảm dần từ trên xuống dưới và từ trái sang phải.

**Bài 17.** Viết hàm sắp xếp ma trận theo đường xoắn ốc từ ngoài vào trong theo chiều kim đồng hồ.

**Bài 18.** Cho biết kết quả của chương trình sau:

```
int main()
{
    int x, *p, *q;
    x = 7 ;
    p = &x;
    q = p;
```

```

    x = x + 5 ;
    printf(“%d%d%d”, x, *p, *q);
    return 0;
}

```

**Bài 19.** Cho biết kết quả của chương trình sau :

```

int main()
{
    int x, *p, *q;
    x = 10 ;
    p = &x;
    q = p;
    *p ++ ;
    *q = *p + x;
    printf(“%d%d%d”, x, *p, *q);
}

```

**Bài 20.** Chương trình sau sai ở dòng lệnh nào? Giải thích ?

```

int main()
{
    int x, *p, *q;
    x = 10 ;      p = &x;
    q = p;        p ++ ;
    *q = *p + x;
    printf(“%d%d%d”, x, *p, *q);
}

```

**Bài 21.** Chương trình sau sai ở dòng lệnh nào? Giải thích?

```

int main()
{
    int x, *p, *q;
    x = 10 ;
    p = &x;
    x = *p ++;
}

```

```

    q ++ ;
    x = *q ++ ;
    x = x + *p ;
    q = &x ;
    *q = *p + x;
    printf(“%d%d%d”, x, *p, *q);
}

```

**Bài 22.** Cho biết kết quả của chương trình sau :

```

void  tinh(int*a, int*b, int&x)
{
    *a = *a + *b ;
    *b = *a + x ;
    x = x + *a + *b ;
}

int main()
{
    int x, y , *q;
    x = 10; y = 20 ;
    p = &x ;
    tinh(&x, &y, p)
    printf(“%d%d%d”, x, y, *p);
}

```

**Bài 23.** Tìm lỗi sai trong đoạn lệnh sau

```

int x[3][12];
int *ptr[12];
ptr = x;

```

**Bài 24.** Viết chương trình khai báo mảng hai chiều có 12x12 phần tử kiểu char. Gán ký tự 'X' cho mọi phần tử của mảng này. Sử dụng con trỏ đến mảng để in giá trị các phần tử mảng lên màn hình ở dạng lưới.

**Bài 25.** Viết khai báo con trỏ func đến một hàm nhận đối số là một số nguyên và trả về giá trị kiểu float.

**Bài 26.** Viết chương trình khai báo mảng 10 con trỏ đến kiểu float, nhận 10 số thực từ bàn phím, sắp xếp lại và in ra màn hình dãy số đã sắp xếp.