

## NHÓM BÀI TẬP SỐ 4

1. Cho mảng A có n phần tử. Viết chương trình minh họa các thao tác cập nhật và xóa phần tử tại vị trí thứ i trong mảng, thực hiện việc chèn thêm một số nguyên x vào sau phần tử thứ i ( $0 \leq i < n$ ).
2. Cho 2 mảng nguyên A, B có thứ tự (tăng dần). Thực hiện việc trộn 2 mảng A và B để tạo ra mảng C cũng có thứ tự (tăng dần).
3. Cho mảng A gồm n phần tử. Viết chương trình tìm số lớn thứ hai trong mảng. Lưu ý, chỉ duyệt qua mảng một lần (dùng một vòng lặp).
4. Cho mảng số nguyên A. Tạo ra 2 mảng nguyên B và C theo quy tắc: mảng B gồm các phần tử dương hoặc bằng 0 của A và mảng C gồm các phần tử âm của A.
5. Viết chương trình nhập vào một mảng, trong quá trình nhập, mảng được sắp xếp thứ tự luôn (tăng / giảm dần).
6. Viết chương trình nhập vào một dãy số A gồm m số thực, nhập vào dãy số B gồm m số thực.
  - a. In ra các phần tử chỉ xuất hiện trong dãy A mà không xuất hiện trong dãy B.
  - b. In ra những phần tử xuất hiện ở cả hai dãy.
7. Hãy liệt kê các số trong mảng 1 chiều các số thực thuộc đoạn  $[x, y]$  cho trước.
8. Hãy liệt kê các số chẵn trong mảng 1 chiều các số nguyên thuộc đoạn  $[x, y]$  cho trước (x, y là các số nguyên).
9. Hãy liệt kê các giá trị trong mảng mà thỏa điều kiện lớn hơn giá trị tuyệt đối của giá trị đứng liền sau nó.
10. Cho mảng 1 chiều các số thực. Hãy viết hàm liệt kê tất cả các giá trị trong mảng có ít nhất 1 lân cận trái dấu với nó.
11. Hãy liệt kê các giá trị trong mảng 1 chiều các số nguyên có chữ số đầu tiên là số chẵn.
12. Tính tổng các phần tử “cực trị” trong mảng. Một phần tử được gọi là cực trị khi nó lớn hơn hoặc nhỏ hơn các phần tử xung quanh nó.
13. Đếm số lượng giá trị lớn nhất có trong mảng một chiều các số thực.
14. Liệt kê tần suất xuất hiện các giá trị trong mảng (Lưu ý: mỗi giá trị liệt kê 1 lần).
15. Cho 2 mảng a, b. Đếm số lượng giá trị chỉ xuất hiện 1 trong 2 mảng.
16. Cho 2 mảng a, b. Liệt kê các giá trị chỉ xuất hiện 1 trong 2 mảng.
17. Kiểm tra mảng số nguyên có tồn tại giá trị 0 hay không? Nếu không tồn tại giá trị 0 thì trả về giá trị 0, ngược lại trả về giá trị 1.
18. Kiểm tra mảng số nguyên có tồn tại 2 giá trị 0 liên tiếp hay không?
19. Kiểm tra mảng các số nguyên có toàn số chẵn không? Nếu có tồn tại giá trị lẻ trả về giá trị 0, ngược lại trả về giá trị 1.
20. Kiểm tra mảng một chiều các số thực có đối xứng không?
21. Ta định nghĩa 1 mảng có tính chất chẵn lẻ, khi tổng của 2 phần tử liên tiếp luôn là số lẻ. Viết hàm kiểm tra mảng có tính chất chẵn lẻ hay không.
22. Hãy cho biết các phần tử trong mảng có lập thành cấp số cộng hay không? Nếu có chỉ ra công sai d.
23. Ta định nghĩa 1 mảng được gọi là dạng sóng, khi phần tử có trị số i lớn hơn hoặc nhỏ hơn 2 phần tử xung quanh. Hãy viết hàm kiểm tra mảng có dạng sóng không.

24. Hãy cho biết tất cả các phần tử trong mảng a có nằm trong mảng b không?
25. Hãy xóa tất cả các số lớn nhất trong mảng các số thực.
26. Hãy xóa tất cả số chính phương trong mảng một chiều các số nguyên.
27. Hãy xóa tất cả các phần tử trùng nhau trong mảng và chỉ giữ lại duy nhất một phần tử.
28. Hãy xóa tất cả các phần tử có tần suất xuất hiện trong mảng lớn hơn 1 lần.
29. Hãy đưa các số chẵn trong mảng về đầu mảng, các số lẻ về cuối mảng và các phần tử 0 nằm ở giữa.
30. Hãy đảo ngược mảng ban đầu.
31. Hãy “dịch trái xoay vòng” các phần tử trong mảng.
32. Hãy “dịch phải xoay vòng” k lần các phần tử trong mảng.
33. Liệt kê tất cả các mảng con có độ dài lớn hơn 2 trong mảng một chiều các số nguyên.
34. Liệt kê các dãy con tăng trong mảng.
35. Tính tổng từng mảng con tăng trong mảng một chiều các số thực.
36. Đếm số lượng mảng con tăng có độ dài lớn hơn 1 trong mảng một chiều các số thực.
37. Cho hai mảng a và b. Hãy cho biết mảng a có phải là mảng con trong mảng b hay không?
38. Cho hai mảng a và b. Hãy đếm số lần xuất hiện của mảng a nằm trong mảng b.
39. Tìm dãy con toàn dương dài nhất trong mảng các số thực.
40. Viết chương trình sắp xếp các phần tử trong một mảng theo thứ tự giảm dần.

1. Cho mảng A có n phần tử. Viết chương trình minh họa các thao tác cập nhật và xóa phần tử tại vị trí thứ i trong mảng, thực hiện việc chèn thêm một số nguyên x vào sau phần tử thứ i ( $0 \leq i < n$ ).

```
#include<stdio.h>
void nhap(int &n, int a[], int &giatri, int &vitri, int &yeucau);
void Loigoi(int x, int vt);
void Capnhat(int n, int a[], int giatri, int vitri);
void Xoa(int n, int a[], int vitri);
void Them(int n, int a[], int giatri, int vitri);
void Xuat(int a);

int main()
{
    int n,a[100],giatri,vitri,yeucau;
    nhap(n,a,giatri,vitri,yeucau);
    if (yeucau == 1)
        Capnhat(n, a, giatri, vitri);
    else if (yeucau == 2)
        Xoa(n, a, vitri);
    else if (yeucau == 3)
        Them(n, a, giatri, vitri);
}
void nhap(int &n, int a[], int &giatri, int &vitri, int &yeucau)
{
    scanf("%d%d%d", &n, &giatri, &vitri); // So phan tu_ Gia tri them
    cap nhap_ vitri xuly_ lua chon yeu cau;
    for (int i = 0; i < n; i++)
        scanf("%d", &a[i]);
    Loigoi(giatri, vitri);
    scanf("%d", &yeucau);
}
```

```

void Loigoi(int x, int vt)
{
    printf("\tVui long lua chon thao tac:\n\t\t1. Cap nhat %d vao phan
tu thu %d.\n\t\t2. Xoa phan tu thu %d.\n\t\t3. Them gia tri %d vao sau
phan tu thu %d.\n", x, vt, vt, x, vt);
}
void Capnhat(int n, int a[], int giatri, int vitri)
{
    a[vitri] = giatri;
    for (int i = 0; i < n; i++)
        Xuat(a[i]);
}

void Xoa(int n, int a[], int vitri)
{
    for (int i = 0; i < n - 1; i++)
    {
        if (i==vitri)
        {
            a[i] = a[i + 1];
            vitri++;
        }
        Xuat(a[i]);
    }
}

void Them(int n, int a[], int giatri, int vitri)
{
    for (int i = n; i > vitri + 1; i--)
        a[i] = a[i-1];
    a[vitri + 1] = giatri;//chen vo gia tri sau la gia tri +1
    for (int i = 0; i < n+1; i++)
        Xuat(a[i]);
}
void Xuat(int a)
{
    printf("%d ", a);
}
}

```

2. Cho 2 mảng nguyên A, B có thứ tự (tăng dần). Thực hiện việc trộn 2 mảng A và B để tạo ra mảng C cũng có thứ tự (tăng dần).

```

#include<stdio.h>
void Kiemtra(int A[], int B[], int C[], int& n, int& m, int& p);
void Xuat(int C[], int p);
void Nhap(int A[], int B[], int& n, int& m); //Nhap 2 mang A B co thu tu tang
dan;

int main()
{
    int A[1000], B[1000], C[1000], n, m, p;
    Nhap(A, B, n, m);
    Kiemtra(A, B, C, n, m, p);
    Xuat(C, p);
}

```

```

}

void Nhap(int A[], int B[], int& n, int& m) //Nhap 2 mang A B co thu tu tang
dan;
{
    scanf_s("%d", &n);
    for (int i = 0; i < n; i++)
        scanf_s("%d", &A[i]);
    scanf_s("%d", &m);
    for (int j = 0; j < m; j++)
        scanf_s("%d", &B[j]);
}

void Kiemtra(int A[], int B[], int C[], int& n, int& m, int& p)
{
    p = 0;
    int i = 0, j = 0;
    while (i < n || j < m)
    {
        if (i < n && j < m)
        {
            if (A[i] <= B[j])
            {
                C[p] = A[i];
                i++;
            }
            else
            {
                C[p] = B[j];
                j++;
            }
            p++;
        }
        else
        {
            if (i >= n && j < m)
            {
                //C[p]=B[j];
                C[p++] = B[j++];
                //p++,j++;
            }
            else if (i < n && j >= m)
            {
                C[p++] = A[i++]; //p++,i++;
            }
        }
    }
}

void Xuat(int C[], int p)
{
    for (int i = 0; i < p; i++)
        printf("%d ", C[i]);
}

```

```

#include<stdio.h>
void tronMang(int A[], int nA, int B[], int nB, int C[], int &nC);
void nhapMang(int A[], int &n);
void xuatMang(int A[], int n);

void main()
{
    int A[100], nA, B[100], nB, C[200], nC;
    nhapMang(A,nA);
    nhapMang(B,nB);
    tronMang(A,nA,B,nB,C,nC);
    xuatMang(C,nC);
}
void tronMang(int A[], int nA, int B[], int nB, int C[], int &nC)
{
    nC=0;
    int iA=0;
    int iB=0;
    while (iA<nA && iB<nB)
        if (A[iA]<B[iB])
            C[nC++]=A[iA++];
        else
            C[nC++]=B[iB++];
    while (iA<nA)//phan du, mang a dai hon
        C[nC++]=A[iA++];
    while (iB<nB)//phan du, mang b dai hon
        C[nC++]=B[iB++];
}
void nhapMang(int A[], int &n)
{
    scanf("%d",&n);
    for (int i=0;i<n;i++)
        scanf("%d",&A[i]);
}
void xuatMang(int A[], int n)
{
    for (int i=0;i<n;i++)
        printf("%d ",A[i]);
}

```

3. Cho mảng A gồm n phần tử. Viết chương trình tìm số lớn thứ hai trong mảng. Lưu ý, chỉ duyệt qua mảng một lần (dùng một vòng lặp).

```

#include<stdio.h>
void Nhap(int A[], int& n );
void Xuly(int A[], int n);
void xuat(int a);

int main()
{
    int A[1000],n;
    Nhap(A,n);
    Xuly(A,n);
}

void Nhap(int A[], int& n )
{

```

```

scanf("%d", &n);
for (int i = 0; i < n; i++)
    scanf("%d", &A[i]);
}
void Xuly(int A[], int n)
{
    int nhat=A[0];
    int nhi ;

    for (int i = 0; i < n; i++)
    {
        if (A[i] > nhat)
        {
            nhi = nhat;
            nhat = A[i];
        }
    }

    xuat(nhi);
}
void xuat(int a)
{
    printf("%d",a);
}

```

4. Cho mảng số nguyên A. Tạo ra 2 mảng nguyên B và C theo quy tắc: mảng B gồm các phần tử dương hoặc bằng 0 của A và mảng C gồm các phần tử âm của A.

```

#include<stdio.h>
void nhap(int &n,int A[]);
void xuly(int n, int A[], int B[], int C[]);
void xuat(int a);
void xuat1(int d);

int main()
{
    int n;
    int A[1000];
    int B[1000];
    int C[1000];
    nhap(n,A);
    xuly(n,A,B,C);
}
void nhap(int &n,int A[])
{
    scanf("%d",&n);
    for(int i=0; i<n; i++)
    {
        scanf("%d",&A[i]);
    }
}

```

```

void xuly(int n, int A[], int B[], int C[])
{
    int i;
    int b = 0;
    int c = 0;
    for(i=0; i<n; i++)
    {
        if(A[i]<=0)
        {
            B[b]=A[i];
            b++;
        }
        else
        {
            C[c]=A[i];
            c++;
        }
    }
    for(int j=0; j<b; j++)
    {
        xuat(B[j]);
    }
    for(int f=0; f<c; f++)
    {
        xuat1(C[f]);
    }
}
void xuat(int a)
{
    printf("%d ",a);
    printf("\n");
}
void xuat1(int d)
{
    printf("%d ",d);
}
}

```

```

#include<stdio.h>
void Nhap(int A[], int& n);
void MangB(int A[], int B[], int n, int& m);
void MangC(int A[], int C[], int n, int& p);
void XuatB(int B[], int m);
void XuatC(int B[], int m);

int main()
{
    int A[1000], B[1000], C[1000], m, n, p;
    Nhap(A, n);
    MangB(A, B, n, m);
    MangC(A, C, n, p);
    XuatC(C, p);
    XuatB(B, m);
}

```

```

void Nhap(int A[], int& n)
{
    scanf_s("%d", &n);
    for (int i = 0; i < n; i++)
        scanf_s("%d", &A[i]);
}

void MangB(int A[], int B[], int n, int &m)
{
    m = 0;
    for (int i = 0; i < n; i++)
        if (A[i] >= 0)
        {
            B[m] = A[i];
            m++;
        }
}

void MangC(int A[], int C[], int n, int &p)
{
    p = 0;
    for (int i = 0; i < n; i++)
        if (A[i] < 0)
        {
            C[p] = A[i];
            p++;
        }
}

void XuatC(int C[], int p)
{
    for (int i = 0; i < p; i++)
        printf("%d ", C[i]);
    printf("\n");
}

void XuatB(int B[], int m)
{
    for (int i = 0; i < m; i++)
        printf("%d ", B[i]);
}

```

```

5
#include<stdio.h>
void nhapMang( int A[], int &n)
{
    printf("Nhap so phan tu cua mang");
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%d",&A[i]);
}
void Hoan_Vi( int &a , int &b)
{
    int temp;
    temp=a;

```



```

        a=b;
        b=temp;
    }
void Sap_Xep( int A[], int n)
{
    for (int i=0;i<n-1;i++)
    {
        for(int j=i+1;j<n;j++)
            {
                if(A[i]>A[j])
                    Hoan_Vi( A[i], A[j] );
            }
    }
}
void xuatMang( int A[], int n)
{
    for(int i=0;i<n;i++)
        printf("%d ",A[i]);
}
int main()
{
    int A[150],n;
    nhapMang(A,n);
    Sap_Xep(A,n);
    xuatMang(A,n);
}
C2
#include <stdio.h>
void XuLy(int A[], int& n);
void Hoan_Vi( int &x, int &y);
void SX_Tang(int A[], int n);
void Xuat(int A[], int n);

int main()
{
    int A[100], n;
    XuLy(A, n);
}

void XuLy(int A[], int& n)
{
    scanf("%d", &n);
    for (int i = 0; i < n; i++)
    {
        scanf("%d", &A[i]);
        SX_Tang(A, i + 1);
        Xuat(A, i+1);
    }
}

void Hoan_Vi( int &x, int &y)
{
    int temp=x;
    x=y;
    y=temp;
}
void SX_Tang(int A[], int n)

```

```

{
    for (int i = 0; i < n - 1; i++)
        for (int j = i + 1; j < n; j++)
            if (A[i] > A[j])
                Hoan_Vi(A[i],A[j]);
}

void Xuat(int A[], int n)
{
    for (int i = 0; i < n; i++)
        printf("%d ", A[i]);
    printf("\n");}

```

```

6.1
#include<stdio.h>
void nhap( int &n, int m[]);
void xuathienamakhongb(int na, int nb, int a[],int b[]);

int main()
{
    int na,nb,a[100],b[100];
    nhap(na,a);
    nhap(nb,b);
    xuathienamakhongb(na,nb,a,b);
}
void nhap( int &n, int m[])
{
    scanf("%d",&n);
    for(int i=0; i<n; i++)
    {
        scanf("%d",&m[i]);
    }
}
void xuathienamakhongb(int na, int nb, int a[],int b[])
{
    for(int i=0; i<na; i++)
    {
        int thuhang=0;
        for(int j=0; j<nb; j++)
        {
            if(a[i]!=b[j])
                thuhang++;
        }
        if(thuhang==nb)
            printf("%d",a[i]);
    }
}

6.2
#include<stdio.h>
void nhap( int &n, int m[]);
void xuathiencahai(int na,int nb, int a[],int b[],int c[]);
int main()
{
    int na,nb,a[100],b[100],c[100];
    nhap(na,a);
    nhap(nb,b);

```

```

        xuathiencahai(na,nb,a,b,c);
}
void nhap( int &n, int m[])
{
    scanf("%d",&n);
    for(int i=0; i<n; i++)
    {
        scanf("%d",&m[i]);
    }
}
void xuathiencahai(int na,int nb, int a[],int b[],int c[])
{
    int p=0;
    for(int i=0; i<na; i++)
    {
        for(int j=0; j<nb; j++)
        {
            if(a[i]==b[j])//1= la gan 2== la gia tri
            {
                c[p]=a[i];
                p++;
            }
        }
    }
    printf("%d ",c[0]);
    for(int h=1; h<p; h++)
    {
        if(c[h]!=c[h-1])
printf("%d",c[h]);
}

```

```

7
#include<stdio.h>
void nhapMang( float A[], int &nA, float &x, float &y);
void Xu_Li( float A[], int nA, float B[], int &nB, float x, float y);
void xuatMang( float B[], int nB);
int main()
{
    float A[100],B[100],x,y;
    int nA,nB;
    nhapMang(A,nA,x,y);
    Xu_Li(A,nA,B,nB,x,y);
    xuatMang(B,nB);
}
void nhapMang( float A[], int &nA, float &x, float &y)
{
    printf("Nhap mang :\n");
    scanf("%d",&nA);
    for(int i=0;i<nA;i++)
        scanf("%f",&A[i]);
    printf("Nhap khoang :\n");
    scanf("%f%f",&x,&y);
}

```

```

void Xu_Li( float A[], int nA, float B[], int &nB, float x, float y)
{
    nB=0;
    for( int i=0;i<nA;i++)
    {
        if( A[i] >= x && A[i] <= y )
            B[nB++]=A[i];
    }
}
void xuatMang( float B[], int nB)
{
    for(int i=0;i<nB;i++)
        printf("%.2f ",B[i]);}

```

```

8
#include<stdio.h>
void nhapMang(int A[], int &nA, int &x ,int &y);
void Xu_Li( int A[], int nA , int B[], int &nB, int  x, int y);
void xuatMang( int B[], int nB);
void main()
{
    int A[100],nA,B[100],nB,x,y;
    nhapMang(A,nA,x,y);
    Xu_Li(A,nA,B,nB,x,y);
    xuatMang(B,nB);
}
void nhapMang(int A[], int &nA, int &x ,int &y)
{
    printf("Nhap so phan tu cua mang");
    scanf("%d",&nA);
    printf("\n");
    printf("Nhap mang A: ");
    for(int i=0;i<nA;i++)
        scanf("%d",&A[i]);
    printf("\n");
    printf(" Nhap khoang x va y ");
    scanf("%d%d",&x,&y);
}
void Xu_Li( int A[], int nA , int B[], int &nB, int  x, int y)
{
    nB=0;
    for(int i=0;i<nA;i++)
    {
        if( A[i]%2==0 )
        {
            if( A[i] >=x && A[i] <= y)
            {
                B[nB++]=A[i];
            }
        }
    }
}
void xuatMang( int B[], int nB)
{
    for(int i=0;i<nB;i++)
        printf("%d ",B[i]);
}

```

```

9
#include<stdio.h>
#include<math.h>
void nhapMang(int A[], int &nA);
void Xu_Li( int A[], int nA, int B[], int &nB);
void xuatMang( int B[], int nB);
void main()
{
    int A[100],nA,B[100],nB;
    nhapMang(A,nA);
    Xu_Li(A,nA,B,nB);
    xuatMang(B,nB);
}
void nhapMang(int A[], int &nA)
{
    printf("Nhap so phan tu cua mang");
    scanf("%d",&nA);
    printf("\n");
    printf("Nhap mang A: ");
    for(int i=0;i<nA;i++)
        scanf("%d",&A[i]);
}
void Xu_Li( int A[], int nA, int B[], int &nB)
{
    nB=0;
    for(int i=0;i<nA-1;i++)
    {
        if( A[i] > abs(A[i+1]) )
        {
            B[nB++]=A[i] ;
        }
    }
}
void xuatMang( int B[], int nB)
{
    for ( int i=0;i<nB;i++)
        printf("%d ",B[i]);}

```

```

10
#include<stdio.h>
void nhapMang(float A[], int &nA);
void Xu_Li( float A[], int nA, float B[], int &nB);
void xuatMang( float B[], int nB);
void main()
{
    float A[100],B[100];
    int nA,nB;
    nhapMang(A,nA);
    Xu_Li(A,nA,B,nB);
    xuatMang(B,nB);
}
void nhapMang(float A[], int &nA)
{
    printf("Nhap so phan tu cua mang");
    scanf("%d",&nA);

```

```

    printf("\n");
    printf("Nhap mang A: ");
    for(int i=0;i<nA;i++)
        scanf("%f",&A[i]);
}
void Xu_Li( float A[], int nA, float B[], int &nB)
{
    nB=0;
    for( int i=1;i<nA-1;i++)
    {
        if( A[i]*A[i-1] <0 || A[i]*A[i+1] <0)
            B[nB++]=A[i];
    }
}
void xuatMang( float B[], int nB)
{
    for(int i=0;i<nB;i++)
        printf("%.2f ",B[i]);
}

```

```

11
#include<stdio.h>
void nhapMang(int A[], int &nA );
void Ket_Qua( int A[], int nA, int B[], int &nB);
void xuatMang( int B[], int nB);
void main()
{
    int A[100],nA,B[100],nB;
    nhapMang(A,nA);
    Ket_Qua(A,nA,B,nB);
    xuatMang(B,nB);
}
void nhapMang(int A[], int &nA )
{
    printf("Nhap so phan tu cua mang");
    scanf("%d",&nA);
    printf("\n");
    printf("Nhap mang A: ");
    for(int i=0;i<nA;i++)
        scanf("%d",&A[i]);
}
bool KT_SoDauTien( int n)
{
    int a;
    while(n!=0)
    {
        a=n;
        if(a/10==0)
        {
            if(a%2==0)
                return true;
            else
                return false;
        }
        n=n/10;
    }
}

```

```

}
void Ket_Qua( int A[], int nA, int B[], int &nB)
{
    nB=0;
    for(int i=0;i<nA;i++)
    {
        if(KT_SoDauTien(A[i]))
            B[nB++]=A[i];
    }
}
void xuatMang( int B[], int nB)
{
    for(int i=0;i<nB;i++)
        printf("%d ",B[i]);
}

```

```

12
#include<stdio.h>
void nhapMang(float A[], int &nA);
float TinhTong( float A[], int nA);
void xuatMang( float A[], int nA);
void xuat( float kq);
int main()
{
    float A[100];
    int nA;
    nhapMang(A,nA);
    float kq=TinhTong(A,nA);
    xuatMang(A,nA);
    xuat(kq);
}
void nhapMang(float A[], int &nA)
{
    printf("Nhap so phan tu cua mang");
    scanf("%d",&nA);
    printf("\n");
    printf("Nhap mang A: ");
    for(int i=0;i<nA;i++)
        scanf("%f",&A[i]);
}
float Tong_CucDai( float A[], int nA)//lon hon cac phan tu lan can
{
    float t1=0;
    for( int i=1;i<nA-1;i++)
    {
        if( A[i] > A[i-1] && A[i] > A[i+1])
            t1=t1+A[i];
    }
    return t1;
}
float Tong_CucTieu( float A[], int nA)
{
    float t2=0;
    for( int i=1;i<nA-1;i++)
    {
        if( A[i]< A[i-1] && A[i]< A[i+1])//be hon cac phan tu lan can

```

```

        t2=t2+A[i];
    }
    return t2;
}
float TinhTong( float A[], int nA)
{
    float a=Tong_CucDai(A,nA);
    float b=Tong_CucTieu(A,nA);
    float t=0;
    t=a+b;
    return t;
}
void xuatMang( float A[], int nA)
{
    for(int i=0;i<nA;i++)
        printf("%.2f ",A[i]);
    printf("\n");
}
void xuat( float kq)
{
    printf("%.2f",kq);
}

```

```

13
#include<stdio.h>
void nhapMang(float A[], int &nA);
int Dem( float A[], int nA);
void xuat( int kq);
void main()
{
    float A[100];
    int nA;
    nhapMang(A,nA);
    int kq=Dem(A,nA);
    xuat(kq);
}
void nhapMang(float A[], int &nA)
{
    printf("Nhap so phan tu cua mang");
    scanf("%d",&nA);
    printf("\n");
    printf("Nhap mang A: ");
    for(int i=0;i<nA;i++)
        scanf("%f",&A[i]);
}
float Tim_Max( float A[], int nA)
{
    float max_=A[0];
    for( int i=1;i<nA;i++)
    {
        if(A[i]> max_)
            max_=A[i];
    }
    return max_;
}
int Dem( float A[], int nA)
{

```



```

float max=Tim_Max(A,nA);
int dem=0;
for(int i=0;i<nA;i++)
{
    if( max==A[i] )
    {
        dem=dem+1;
    }
}
return dem;
}
void xuat( int kq)
{
    printf("%d",kq);
}

```

14

// y tuong: sap xep mang a loai phan tu trung roi dung ham tim kiem

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

```
void nhapMang( int A[], int &nA);
```

```
void Sap_Xep_Tang_Dan(int A[], int nA);
```

```
void Do_Mang( int A[], int nA, int B[] , int &nB);
```

```
void Xoa_Trung( int B[], int &nB);
```

```
void Liet_Ke(int A[], int nA, int B[],int nB);
```

```
int main()
```

```
{
```

```
    int A[100], nA,B[100],nB;
```

```
    nhapMang(A,nA);
```

```
    Sap_Xep_Tang_Dan(A,nA);
```

```
    Do_Mang(A,nA,B,nB);
```

```
    Xoa_Trung(B,nB);
```

```
    Liet_Ke( A, nA,B,nB);
```

```
}
```

```
void nhapMang( int A[], int &nA)
```

```
{
```

```
    printf("Nhap so phan tu cua mang");
```

```
    scanf("%d",&nA);
```

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

```
    printf("Nhap mang A: ");
```

```
    for(int i=0;i<nA;i++)
        scanf("%d",&A[i]);
```

```
}
```

```
void Hoan_Vi(int &x, int &y)
```

```
{
```

```
    int temp=x;
```

```
    x=y;
```

```
    y=temp;
```

```
}
```

```
void Sap_Xep_Tang_Dan(int A[], int nA)
```

```
{
```

```
    for(int i=0;i<nA;i++)// for(int i=0;i<nA-1;i++)
```

```
    {
```

```
        for(int j=i+1;j<nA;j++)
```

```
        {
```

```
            if(A[i]>A[j])
```

```
                Hoan_Vi( A[i], A[j] );
```

```

    }
}
((void Do_Mang( int A[], int nA, int B[] , int &nB)
{
    nB=0;
    for ( int i=0;i<nA;i++)
    {
        B[nB++]=A[i];
    }
}
void Xoa_Trung( int B[], int &nB)
{
    for(int i=0;i<nB;i++)
    {
        if(B[i]==B[i+1])
        {
            for(int j=i;j<nB;j++)
            {
                B[j]=B[j+1];
            }
            nB=nB-1;
            i=i-1;
        }
    }
}))
void Do_Mang( int A[], int nA, int B[] , int &nB)
{
    nB=0;
    for ( int i=0;i<nA;i++)
    {
        if(A[i]!=A[i-1])
        {
            B[nB]=A[i];
            printf("%d ",B[nB]);
            nB++;
        }
    }
}
void Liet_Ke(int A[], int nA, int B[],int nB)
{
    for( int i=0;i<nB;i++)
    {
        int dem=0;
        for(int j=0;j<nA;j++)
        {
            if( B[i] == A[j] )
            {
                dem=dem+1;
            }
        }
        printf(" \n Gia tri: %d \n Xuat hien so lan la: %d \n",B[i],dem);
    }
}

```

```

#include<stdio.h>
void nhap( int &n, int m[]);
void xuathienamakhongb(int na, int nb, int a[],int b[]);

int main()
{
    int na,nb,a[100],b[100];
    nhap(na,a);
    nhap(nb,b);
    xuathienamakhongb(na,nb,a,b);
}
void nhap( int &n, int m[])
{
    scanf("%d",&n);
    for(int i=0; i<n; i++)
    {
        scanf("%d",&m[i]);
    }
}
void xuathienamakhongb(int na, int nb, int a[],int b[])
{
    int dem=0;
    for(int i=0; i<na; i++)
    {
        int thuhang=0;
        for(int j=0; j<nb; j++)
        {
            if(a[i]!=b[j])
                thuhang++;
        }
        if(thuhang==nb)
            dem++;
    }
    for(int h=0; h<nb; h++)
    {
        int thuhang1=0;
        for(int g=0; g<na; g++)
        {
            if(b[h]!=a[g])
                thuhang1++;
        }
        if(thuhang1==nb)
            dem++;
    }
    printf("%d",dem);
}

```

16

```

#include<stdio.h>
void nhap( int &n, int m[]);
void xuathienamakhongb(int na, int nb, int a[],int b[]);

int main()
{
    int na,nb,a[100],b[100];

```

```

        nhap(na,a);
        nhap(nb,b);
        xuathienamakhongb(na,nb,a,b);
    }
void nhap( int &n, int m[])
{
    scanf("%d",&n);
    for(int i=0; i<n; i++)
    {
        scanf("%d",&m[i]);
    }
}
void xuathienamakhongb(int na, int nb, int a[],int b[])
{
    int dem=0;
    for(int i=0; i<na; i++)
    {
        int thuhang=0;
        for(int j=0; j<nb; j++)
        {
            if(a[i]!=b[j])
                thuhang++;
        }
        if(thuhang==nb)
            printf("%d ",a[i]);
    }
    for(int h=0; h<nb; h++)
    {
        int thuhang1=0;
        for(int g=0; g<na; g++)
        {
            if(b[h]!=a[g])
                thuhang1++;
        }
        if(thuhang1==nb)
            printf("%d ",b[h]);
    }
}
}

```

```

17
#include<stdio.h>
void nhapMang(int A[],int &nA);
int Xu_Li( int A[],int nA);
void xuat( int kq);
void main()
{
    int A[100],nA;
    nhapMang(A,nA);
    int kq=Xu_Li(A,nA);
    xuat(kq);
}
void nhapMang(int A[],int &nA)
{
    printf("Mang A: ");
    scanf("%d",&nA);
    for(int i=0;i<nA;i++)
        scanf("%d",&A[i]);
}

```

```

}
int Xu_Li( int A[],int nA)
{
    int flag,dem=0;
    for(int i=0;i<nA;i++)
    {
        flag=0;
        if(A[i] ==0)
        {
            flag=1;
            break;
        }
    }
    if(flag==1)
        return 1;
    else
        return 0;
}
void xuat( int kq)
{
    if(kq==1)
        printf("1");
    else
        printf("0");
}

```

```

18
#include<stdio.h>
void nhapMang(int A[],int &nA);
int Xu_Li( int A[],int nA);
void xuat( int kq);
void main()
{
    int A[100],nA;
    nhapMang(A,nA);
    int kq=Xu_Li(A,nA);
    xuat(kq);
}
void nhapMang(int A[],int &nA)
{
    printf("Mang A: ");
    scanf("%d",&nA);
    for(int i=0;i<nA;i++)
        scanf("%d",&A[i]);
}
int Xu_Li( int A[],int nA)
{
    int flag,dem=0;
    for(int i=0;i<nA-1;i++)
    {
        flag=0;
        if(A[i] ==0 && A[i+1]==0)
        {
            flag=1;
            break;
        }
    }
}

```

```

    }
    if(flag==1)
        return 1;
    else
        return 0;
}
void xuat( int kq)
{
    if(kq==1)
        printf("Co");
    else
        printf("0");
}

```

```

19
#include<stdio.h>
void nhapMang(int A[],int &nA);
int Xu_Li( int A[],int nA);
void xuat( int kq);
void main()
{
    int A[100],nA;
    nhapMang(A,nA);
    int kq=Xu_Li(A,nA);
    xuat(kq);
}
void nhapMang(int A[],int &nA)
{
    printf("Mang A: ");
    scanf("%d",&nA);
    for(int i=0;i<nA;i++)
        scanf("%d",&A[i]);
}
int Xu_Li( int A[],int nA)
{
    int flag,dem=0;
    for(int i=0;i<nA;i++)
    {
        flag=1;
        if(A[i] %2!=0)
        {
            flag=0;
            break;
        }
    }
    if(flag==1)
        return 1;
    else
        return 0;
}
void xuat( int kq)
{
    if(kq==1)
        printf("1");
    else
        printf("0");
}

```

```
}
```

```
20
```

```
#include<stdio.h>
void nhapMang(int A[],int &nA);
int KT( int A[], int nA);
void xuat( int kq);
void main()
{
    int A[100],nA;
    nhapMang(A,nA);
    int kq=KT(A,nA);
    xuat(kq);
}
void nhapMang(int A[],int &nA)
{
    printf("Mang A: ");
    scanf("%d",&nA);
    for(int i=0;i<nA;i++)
        scanf("%d",&A[i]);
}
int KT( int A[], int nA)
{
    int flag;
    for(int i=0;i<nA;i++)
    {
        flag=1;
        if(A[i] != A[nA-i-1])
        {
            flag=0;
            break;
        }
    }
    if(flag==1)
        return 1;
    else
        return 0;
}
void xuat( int kq)
{
    if(kq==1)
        printf("1");
    else
        printf("0");
}
}
```

```
21
```

```
#include<stdio.h>
void nhapMang( int A[], int &nA);
int KT( int A[] , int nA);
void xuat( int kq);
void main()
{
    int A[100],nA;
    nhapMang(A,nA);
    int kq=KT(A,nA);
}
```

```

        xuat(kq);
    }
void nhapMang( int A[], int &nA)
{
    scanf("%d",&nA);
    for( int i=0;i<nA;i++)
        scanf("%d",&A[i]);
}
int KT( int A[] , int nA)
{
    int flag;
    for( int i=0;i<nA-1;i++)
    {
        flag=1;
        if( (A[i] + A[i+1])%2 !=0 )
        {
            flag=0;
            break;
        }
    }
    if( flag==1)
        return 1;
    else
        return 0;
}
void xuat( int kq)
{
    if(kq==1)
        printf("Chan\n");
    else
        printf("Le\n");
}

```

```

22
#include<stdio.h>
void nhap( int A[] , int &nA);
int Tinhd( int A[] , int nA , int &d);
void xuat(int kq,int d);
void main()
{
    int A[100],nA,d;
    nhap(A,nA);
    int kq=Tinhd(A,nA,d);
    xuat(kq,d);
}
void nhap( int A[] , int &nA)
{
    scanf("%d",&nA);
    for(int i=0;i<nA;i++)
        scanf("%d",&A[i]);
}
int Tinhd( int A[] , int nA,int &d)
{
    int flag;
    for(int i=0;i<nA-2;i++)
    {

```



```

        flag=1;
        if( (A[i]+A[i+2]) != 2*A[i+1] )
        {
            flag=0;
            break;
        }
    }
    if( flag==1)
    {
        d=A[2]-A[1];
        return 1;
    }
    else
        return 0;
}
void xuat(int kq,int d)
{
    if(kq==1)
        printf("%d\n",d);
    else
        printf("0\n");
}

```

23

```

#include<stdio.h>
void nhap( int A[], int &nA);
int KT( int A[], int nA);
void xuat( int kq);
int main()
{
    int A[100],nA;
    nhap(A,nA);
    int kq=KT(A,nA);
    xuat(kq);
}
void nhap( int A[], int &nA)
{
    scanf("%d",&nA);
    for(int i=0;i<nA;i++)
        scanf("%d",&A[i]);
}
int KT( int A[], int nA)
{
    int dem=1;
    for( int i=1;i<nA-1;i++)
    {
        if((A[i]>A[i-1] && A[i]>A[i+1] || A[i]<A[i-1] && A[i]<A[i+1]))
        {
            dem++;
        }
    }
    if(dem==nA-1)
    {
        return 1;
    }
    return 0;
}

```

```

    }
void xuat( int kq)
{
    if(kq==1)
        printf("co\n");
    if(kq==0)
        printf("ko\n");
}

```

```

24
#include<stdio.h>
void nhapMang( int A[],int &nA, int B[],int &nB);

int KT(int A[], int nA, int B[], int nB);
void xuat( int kq);
int main()
{
    int A[100],nA,B[100],nB;
    nhapMang(A,nA,B,nB);
    int kq=KT( A,nA,B,nB);
    xuat(kq);
}
void nhapMang( int A[],int &nA, int B[],int &nB)
{
    scanf("%d",&nA);
    for(int i=0;i<nA;i++)
        scanf("%d",&A[i]);
    printf("\n");
    scanf("%d",&nB);
    for(int i=0;i<nB;i++)
        scanf("%d",&B[i]);
}

int KT(int A[], int nA, int B[], int nB)
{
    int co=0;
    for( int i=0;i<nA;i++)
    {
        int dem=0;
        for( int j=0;j<nB;j++)
        {
            if(A[i] == B[j])
            {
                dem=dem+1;
            }
        }
        if (dem>0)
            co++;
    }
    if(co==nA)
        return 1;
    else
        return 0;
}
void xuat( int kq)

```

```

{
    if( kq==1)
        printf("co\n");
    else
        printf(" ko \n");
}

```

25

```

#include<stdio.h>
void nhapMang( float A[], int &nA);
void Xoa( float A[], int &nA);
void xuatMang( float A[] , int nA);
int main()
{
    float A[100];
    int nA;
    nhapMang(A,nA);
    Xoa(A,nA);
    xuatMang(A,nA);
    //printf("%d",nA);
}
void nhapMang( float A[], int &nA)
{
    scanf("%d",&nA);
    for(int i=0;i<nA;i++)
        scanf("%f",&A[i]);
}
float Tim_Max( float A[], int nA)
{
    float max=A[0];
    for( int i=1;i<nA;i++)
    {
        if(A[i]>max)
            max=A[i];
    }
    return max;
}
void Xoa( float A[], int &nA)
{
    float max_x=Tim_Max(A,nA);
    for(int i=0;i<nA;i++)
    {
        if(A[i]==max_x)
        {
            for(int j=i;j<nA;j++)
            {
                A[j]=A[j+1];
            }
            nA=nA-1;
            i=i-1;//giu nguyen vi tri i;
        }
    }
}
//cach hai tao mang moi dua mang cu vao neu gap max thi bo qua sau do xuat mang moi
void xuatMang( float A[] , int nA)

```

```
{
    for(int i=0;i<nA;i++)
        printf("%.2f ",A[i]);
}
```

```
26
#include<stdio.h>
#include<math.h>
void nhapMang( int A[] , int &nA);
void Xu_Li( int A[], int &nA);
void xuatMang( int A[], int nA);
void main()
{
    int A[150],nA;
    nhapMang(A,nA);
    Xu_Li(A,nA);
    xuatMang(A,nA);
}
void nhapMang( int A[] , int &nA)
{
    scanf("%d",&nA);
    for( int i=0; i<nA;i++)
        scanf("%d",&A[i]);
}
bool SoChinhPhuong( int x)
{
    int a,b;
    a=int(sqrt(x));
    b=a*a;
    if(b==x)
        return true;
    else
        return false;
}
void Xu_Li( int A[], int &nA)
{
    for(int i=0;i<nA;i++)
    {
        if(SoChinhPhuong(A[i]))
        {
            for( int j=i;j<nA;j++)
            {
                A[j]=A[j+1];
            }
            nA=nA-1;
            i=i-1;
        }
    }
}
void xuatMang( int A[], int nA)
{
    for( int i=0; i<nA;i++)
        printf("%d ",A[i]);
}
```

```
27
#include<stdio.h>
void nhapMang( int A[] , int &nA);
```

```

void Sap_Xep( int A[], int nA);
void Xu_Li( int A[], int &nA);

int main()
{
    int A[150],nA;
    nhapMang(A,nA);
    Sap_Xep(A,nA);
    Xu_Li(A,nA);
}
void nhapMang( int A[] , int &nA)
{
    scanf("%d",&nA);
    for( int i=0; i<nA;i++)
        scanf("%d",&A[i]);
}
void Hoan_Vi( int &x, int &y)
{
    int temp=x;
    x=y;
    y=temp;
}
void Sap_Xep( int A[], int nA)
{
    for(int i=0;i<nA;i++)
        {
            for(int j=i+1;j<nA;j++)
                {
                    if(A[i]>A[j])
                        Hoan_Vi( A[i], A[j] );
                }
        }
}
void Xu_Li( int A[], int &nA)
{
    for(int i=0;i<nA;i++)//for(int i=0;i<nA-1;i++)
    {
        if(A[i]==A[i+1])
        {
            for( int j=i;j<nA;j++)
            {
                A[j]=A[j+1];
            }
            nA=nA-1;
            i=i-1;//giu vi tri
        }
    }for( int i=0; i<nA;i++)
        printf("%d ",A[i]);
}

```

28

```

#include<stdio.h>
void nhapMang( int A[] , int &nA);
void Sap_Xep( int A[], int nA);
void Xu_Li( int A[], int &nA );
void xuatMang(int A[] , int nA);

```



```

        }
    }
}
void xuatMang(int A[] , int nA)
{
    for( int i=0; i<nA;i++)
        printf("%d ",A[i]);
}

```

```

29
#include<stdio.h>
void nhapMang( int A[] , int &nA);
void Xu_Li( int A[], int nA, int B[], int &nB);
void xuatMang(int B[] , int nB);
int main()
{
    int A[150],nA,B[150],nB;
    nhapMang(A,nA);
    Xu_Li(A,nA,B,nB);
    xuatMang(B,nB);
}
void nhapMang( int A[] , int &nA)
{
    scanf("%d",&nA);
    for( int i=0; i<nA;i++)
        scanf("%d",&A[i]);
}
void Xu_Li( int A[], int nA, int B[], int &nB)
{
    nB=0;
    for( int i=0;i<nA;i++)//chay vong lap chan
    {
        if( A[i]%2 == 0 && A[i]!=0 )
        {
            B[nB++]=A[i];
        }
    }

    for( int i=0;i<nA;i++)//chay 0
    {
        if(A[i]==0)
        {
            B[nB++]=A[i];
        }
    }

    for( int i=0;i<nA;i++)//chay vong lap le
    {
        if( A[i]%2!=0 )
        {
            B[nB++]=A[i];
        }
    }
}
void xuatMang(int B[] , int nB)

```

```

{
    for( int i=0; i<nB;i++)
        printf("%d ",B[i]);
}

```

```

30
#include<stdio.h>
void nhapMang(int A[], int &nA);
void Xu_Li(int A[], int nA);

int main()
{
    int A[100],nA;
    nhapMang(A,nA);
    Xu_Li(A,nA);
}

void nhapMang( int A[] , int &nA)
{
    scanf("%d",&nA);
    for( int i=0; i<nA;i++)
        scanf("%d",&A[i]);
}

void Xu_Li(int A[], int nA)
{
    int B[100];
    int i;
    int j;
    for(i=0, j=nA-1; i<nA , j>=0; i++, j--)
    {
        B[j]=A[i];
    }
    for(int h=0; h<nA; h++)
        printf("%d ",B[h]);
}

```

```

31
#include<stdio.h>
void nhapMang( int A[], int &nA, int &t ,int &p);
void Xu_Li_1( int A[], int nA, int B[], int &nB, int t);
void Xu_Li_2( int A[], int nA, int C[], int &nC, int p);
void xuatMang( int B[], int nB, int C[], int nC);
void main()
{
    int A[100],nA,B[150],nB,C[150],nC,t,p;
    nhapMang( A,nA,t,p);
    Xu_Li_1(A,nA,B,nB,t);
    Xu_Li_2(A,nA,C,nC,p);
    xuatMang(B,nB,C,nC);
}

void nhapMang( int A[], int &nA, int &t ,int &p)
{
    scanf("%d",&nA);
    for(int i=0;i<nA;i++)
        scanf("%d",&A[i]);
    printf("Trai may lan");
}

```



```

scanf("%d",&t);
printf("Phai may lan");
scanf("%d",&p);
}
void Xu_Li_1( int A[], int nA, int B[], int &nB, int t)
{
    if(t>nA)
        t=t-nA;
    nB=0;
    for( int i=t;i<nA;i++)
    {
        B[nB++]=A[i];
    }
    for(i=0;i<t;i++)
    {
        B[nB++]=A[i];
    }
}
void Xu_Li_2( int A[], int nA, int C[], int &nC, int p)
{
    if(p>nA)
        p=p-nA;
    if(p<nA)
        p=nA-p;
    nC=0;
    for( int i=p;i<nA;i++)
    {
        C[nC++]=A[i];
    }
    for(i=0;i<p;i++)
    {
        C[nC++]=A[i];
    }
}
void xuatMang( int B[], int nB, int C[], int nC)
{
    for(int i=0;i<nB;i++)
        printf("%d ",B[i]);
    printf("\n");
    for(i=0;i<nC;i++)
        printf("%d ",C[i]);
}

```

32

```

33
#include<stdio.h>
void nhapMang( int A[], int &nA);
void Xu_Li( int A[] , int nA);
void main()
{
    int A[100],nA;
    nhapMang(A,nA);
    Xu_Li(A,nA);
}
void nhapMang( int A[], int &nA)
{

```

```

scanf("%d",&nA);
for( int i=0;i<nA;i++)
    scanf("%d",&A[i]);
}
void Xu_Li( int A[] , int nA)
{
    int Dodai;
    for(int i=0;i<nA;i++)
    {
        for( int Dodai=1+i;Dodai<=nA;Dodai++)
        {
            for( int j=i;j<Dodai;j++)
            {
                printf("%d ",A[j]);
            }
            printf("\n");
        }
    }
}

```

34

35

36

37

38

```

#include<stdio.h>
void nhapMang( int A[], int &nA , int B[] ,int &nB);
int Kiem_Tra( int A[], int nA, int B[] , int nB, int &dem);
void xuat( int kq, int dem);
void main()
{
    int A[150],nA,B[150],nB,dem;
    nhapMang(A,nA,B,nB);
    int kq=Kiem_Tra( A, nA, B , nB,dem);
    xuat(kq,dem);
}
void nhapMang( int A[], int &nA , int B[] ,int &nB)
{
    scanf("%d",&nA);
    for( int i=0;i<nA;i++)
        scanf("%d",&A[i]);
    scanf("%d",&nB);
    for( i=0;i<nB;i++)
        scanf("%d",&B[i]);
}
int Kiem_Tra( int A[], int nA, int B[] , int nB, int &dem)
{
    int flag=0,timkiem;
    dem=0;
    for(int i=0;i<nB;i++)
    {
        if(B[i]==A[0])
        {
            timkiem=i;
            flag=1;
            for( int j=0;j<nA;j++)

```

```

        {
            if(A[j]!=B[timkiem++])
            {
                flag=0;
                break;
            }
        }
        if(flag==1)
        {
            dem=dem+1;
        }
    }
}
if(dem==0)
    return 0;
else
    return 1;
}
void xuat( int kq, int dem)
{
    if(kq==0)
        printf("ko co");
    else
        printf("%d",dem);
}

```

```

39
#include<stdio.h>
void main()
{
}
void DayConToanDuongDaiNhat(float a[MAX],int n)
{
    int i,l,k,test,vt,pt=0;
    for (i=0;i<n;i++)
    {
        for (l=i; l<=n;l++)
        {
            for (test = 1,k=0; k<l ; k++)
            {
                if (a[k]<0)
                {
                    test = 0; break;
                }
            }
            if (test == 1)
            {
                if (pt<k)
                {
                    pt=k;
                    vt=i;
                }
            }
        }
    }
}
for (vt; vt<pt; vt++)

```

```
        printf("%16f",a[vt]);
    }
40
```

**Bài 1:**

Nhập vào 2 số nguyên dương m và y. Cho biết tháng m của năm y có bao nhiêu ngày.

*Đầu vào:* Hai số nguyên dương m và y ( $1 \leq m \leq 12$ ,  $1000 \leq y \leq 3000$ ).

*Đầu ra:* Một số nguyên dương duy nhất cho biết số ngày trong tháng m của năm y.

Ví dụ:

INPUT	OUTPUT
1 1999	31
2 2000	29
11 2020	30

**Bài 2:**

Nhập vào ba số nguyên h, m, s lần lượt là giờ, phút, giây. Cho biết sau đó một giây là mấy giờ, mấy phút, mấy giây.

*Đầu vào:* Ba số nguyên h, m, s ( $0 \leq h < 24$ ,  $0 \leq m, s < 60$ ).

*Đầu ra:* Ba số nguyên theo thứ tự là giờ, phút, giây, biểu thị kết quả theo yêu cầu bài toán, mỗi số cách nhau bởi một khoảng trắng.

Ví dụ:

INPUT	OUTPUT
19 20 0	19 20 1
8 10 59	8 11 0

**Bài 3:**

Nhập vào ba số nguyên d, m, y lần lượt là ngày, tháng, năm. Cho biết sau đó một ngày là ngày nào.

*Đầu vào:* Ba số nguyên d, m, y ( $1 \leq d \leq 31$ ,  $1 \leq m \leq 12$ ,  $1000 \leq y \leq 3000$ ).

*Đầu ra:* Ba số nguyên theo thứ tự là ngày, tháng, năm, biểu thị kết quả theo yêu cầu bài toán, mỗi số cách nhau bởi một khoảng trắng.

Ví dụ:

INPUT	OUTPUT
31 1 2015	1 2 2015
12 12 2015	13 12 2015
28 2 2016	29 2 2016

```

1
#include <stdio.h>
void nhap(int& m, int& y);
int xuLy(int m, int y);
void xuat(int kq);
void main()
{
    int m, y;
    nhap(m, y);
    int kq = xuLy(m, y);
    xuat(kq);
}

```

```

}
void nhap(int& m, int& y)
{
    scanf_s("%d%d", &m, &y);
}
int xuLy(int m, int y)
{
    if (m == 1 || m == 3 || m == 5 || m == 7 || m == 8 || m == 10 || m == 12)
        return 31;
    if (m == 4 || m == 6 || m == 9 || m == 11)
        return 30;
    if (y % 400 == 0 || (y % 4 == 0 && y % 100 != 0))
        return 29;
    return 28;
}
void xuat(int kq)
{
    printf("%d", kq);
}

```

```

2
#include <stdio.h>
void nhap(int& h, int& m, int& s);
void xuLy(int& h, int& m, int& s);
void xuat(int h, int m, int s);
void main()
{
    int h, m, s;
    nhap(h, m, s);
    xuLy(h, m, s);
    xuat(h, m, s);
}
void nhap(int& h, int& m, int& s)
{
    scanf_s("%d%d%d", &h, &m, &s);
}
void xuLy(int& h, int& m, int& s)
{
    s += 1;
    if (s == 60)
    {
        s = 0;
        m += 1;
        if (m == 60)
        {
            m = 0;
            h += 1;
            if (h == 24)
            {
                h = 0;
            }
        }
    }
}
void xuat(int h, int m, int s)
{
    printf("%d %d %d", h, m, s);
}

```

```

#include <stdio.h>
void nhap(int& d, int& m, int& y);
int ngayTrongThang(int m, int y);
void xuLy(int& d, int& m, int& y);
void xuat(int d, int m, int y);
void main()
{
    int d, m, y;
    nhap(d, m, y);
    xuLy(d, m, y);
    xuat(d, m, y);
}
void nhap(int& d, int& m, int& y)
{
    scanf_s("%d%d%d", &d, &m, &y);
}
int ngayTrongThang(int m, int y)
{
    if (m == 1 || m == 3 || m == 5 || m == 7 || m == 8 || m == 10 || m == 12)
        return 31;
    if (m == 4 || m == 6 || m == 9 || m == 11)
        return 30;
    if (y % 400 == 0 || (y % 4 == 0 && y % 100 != 0))
        return 29;
    return 28;
}
void xuLy(int& d, int& m, int& y)
{
    d++;
    if (d > ngayTrongThang(m, y))
    {
        d = 1;
        m++;
        if (m == 13)
        {
            m = 1;
            y++;
        }
    }
}
void xuat(int d, int m, int y)
{
    printf("%d %d %d", d, m, y);
}

```

### Bài 1:

Nhập vào một số nguyên dương  $n$ . Tính tổng  $S = 1^2 + 2^2 + \dots + n^2$ .

*Đầu vào:* Một số nguyên dương  $n$ .

*Đầu ra:* Một số duy nhất cho biết kết quả của tổng  $S$ .

Ví dụ:

INPUT	OUTPUT
2	5
5	55
10	385

**Bài 2:**

Nhập vào một số nguyên dương n. Tính tổng các chữ số của n.

*Đầu vào:* Một số nguyên dương n.

*Đầu ra:* Một số duy nhất cho biết kết quả tổng các chữ số của n.

Ví dụ:

INPUT	OUTPUT
10	1
234	9
456	15

**Bài 3:**

Nhập vào một số nguyên dương n. Cho biết có bao nhiêu số nguyên tố nhỏ hơn n.

*Đầu vào:* Một số nguyên dương n.

*Đầu ra:* Một số duy nhất cho biết số các số nguyên tố nhỏ hơn n.

Ví dụ:

INPUT	OUTPUT
5	2
21	8
100	25

```
1
#include <stdio.h>
void nhap(int& n);
int xuLy(int n);
void xuat(int kq);
void main()
{
    int n;
    nhap(n);
    int kq = xuLy(n);
    xuat(kq);
}
void nhap(int& n)
{
    scanf_s("%d", &n);
}
int xuLy(int n)
{
    int tong = 0;
    for (int i = 1; i <= n; i++)
    {
        tong += i * i;
    }
    return tong;
}
void xuat(int kq)
{
    printf("%d", kq);
}
```

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

```

void nhap(int& n);
int xuLy(int n);
void xuat(int kq);
void main()
{
    int n;
    nhap(n);
    int kq = xuLy(n);
    xuat(kq);
}
void nhap(int& n)
{
    scanf_s("%d", &n);
}
int xuLy(int n)
{
    int tong = 0;
    while (n != 0)
    {
        tong += n % 10;
        n /= 10;
    }
    return tong;
}
void xuat(int kq)
{
    printf("%d", kq);
}

#include <stdio.h>
#include <math.h>
void nhap(int& n);
int laNguyenTo(int n);
int xuLy(int n);
void xuat(int kq);
void main()
{
    int n;
    nhap(n);
    int kq = xuLy(n);
    xuat(kq);
}
void nhap(int& n)
{
    scanf_s("%d", &n);
}
int laNguyenTo(int n)
{
    if (n < 2)
        return 0;
    for (int i = 2; i <= sqrt(n); i++)
        if (n % i == 0)
            return 0;
    return 1;
}
int xuLy(int n)
{
    int dem = 0;
    for (int i = 2; i < n; i++)
        if (laNguyenTo(i))

```



```

        dem++;
        return dem;
    }
void xuat(int kq)
{
    printf("%d", kq);
}

#include<stdio.h>
#include<math.h>
void nhap(int &n, float &x);
float giaithua (int n);
float tinh(int n, float x);
void xuat(float kq);
int main()
{
    int n;
    float x;
    nhap(n, x);
    float kq = tinh(n, x);
    xuat(kq);
}

void nhap(int &n, float &x)
{
    scanf("%d%f",&n,&x);
}
float giaithua (int n)
{
    float giaithua=1;
    for(int i = 1 ; i <= n ; i ++)
    {
        giaithua=giaithua*i;
    }
    return giaithua;
}
float tinh(int n, float x)
{
    float sum=1;
    float s=0;
    for (int i=1;i<=n;i++)
    {
        s=pow(x,i)/giaithua(i);
        sum+=s;
    }
    return sum;
}

void xuat(float kq)
{
    printf("%.2f",kq);
}

```

HK2 2013-2014

```

Bai 2
#include<stdio.h>
void nhap(int a[]);
void tienhuong(int a[], int b[]);

```

```

void xuat(int a);
int main()
{
    int a[6],b[6];
    nhap(a);
    nhap(b);
    tienthuong(a,b);
}
void nhap(int a[])
{
    for(int i=0; i<6;i++)
    {
        scanf("%d",&a[i]);
    }
}
void tienthuong(int a[], int b[])
{
    int dem=0;
    int tongsocuai;
    for(int i=5; i>=0; i--)
    {
        if(a[i]==b[i])
        {
            dem++;
            tongsocuai+=i;
        }
    }
    if(dem==6)
    xuat(10000000);
    else if(dem==5)
    {
        if(tongsocuai==15)//5+4+3+2+1,5 so cuoi
            xuat(5000000);
        else
            xuat(4000000);
    }
    else if(tongsocuai==14)
        xuat(2000000);
    else if(tongsocuai==12)
        xuat(1000000);
    else if(tongsocuai==9)
        xuat(500000);
    else
        xuat(0);
}
void xuat(int a)
{
    printf("%d",a);
}

```

```

Bai 1
#include<stdio.h>
void nhap(int &n, float a[]);
void xuly(int n, float a[]);

int main()
{
    int n;
    float a[50];
    nhap(n,a);
    xuly(n,a);
}
void nhap(int &n, float a[])
{
    scanf("%d",&n);
    for(int i=0; i<n; i++)
    {
        scanf("%f",&a[i]);
    }
}
void xuly(int n, float a[])
{
    int b[999];
    b[0]=1;

    for(int i=0; i<999; i++)
    {
        b[i+1]=b[i]+1;
    }
    int c[100];
    int p=0;
    for(int j=0; j<999; j++)
    {
        int dem=0;
        for(int f=0; f<n ;f++)
        {
            if(b[j]>=a[f] && b[j]+a[f]>=0)
                dem++;
        }
        if (dem==n)
            c[p++]=b[j];
    }
    int min=c[0];
    for(int k=0; k<p; k++)
    {
        if(c[k]<min)
            min=c[k];
    }
    printf("%d",min);
}

```

