

NHÓM BÀI TẬP SỐ 3

1. Nhập vào một mảng có n số nguyên. Xuất ra mảng vừa nhập.
2. Nhập vào mảng có n số thực. Tính tổng các phần tử trong mảng.
3. Viết chương trình tìm số lớn nhất trong một mảng số nguyên.
4. Viết chương trình tính trung bình cộng của một mảng n số nguyên.
5. Viết chương trình tính trung bình cộng của các số chia hết cho 5 trong mảng.
6. Viết chương trình đếm số nguyên tố có trong một mảng các số nguyên.
7. Viết chương trình đếm số lượng số hoàn thiện trong một mảng.
8. Viết chương trình tính trung bình cộng của các số nguyên tố trong mảng.
9. Tìm giá trị âm đầu tiên trong mảng một chiều các số âm thực. Nếu mảng không có giá trị âm thì trả về giá trị không âm là giá trị 1.
10. Viết hàm tìm “số chẵn đầu tiên” trong mảng các số nguyên. Nếu mảng không có giá trị chẵn thì hàm sẽ trả về giá trị không chẵn là -1.
11. Tìm “số nguyên tố đầu tiên” trong mảng một chiều các số nguyên. Nếu mảng không có số nguyên tố thì trả về giá trị -1.
12. Tìm “số hoàn thiện đầu tiên” trong mảng một chiều các số nguyên. Nếu mảng không có số hoàn thiện thì trả về giá trị -1.
13. Tìm “số nguyên dương cuối cùng” trong mảng số thực. Nếu mảng không có giá trị dương thì trả về giá trị -1.
14. Tìm “số nguyên tố cuối cùng” trong mảng một chiều các số nguyên. Nếu mảng không có số nguyên tố thì trả về giá trị -1.
15. Tìm “số hoàn thiện cuối cùng” trong mảng một chiều các số nguyên. Nếu mảng không có số hoàn thiện thì trả về giá trị 0.
16. Viết chương trình tìm số âm lớn nhất trong một mảng các số nguyên.
17. Tìm số “âm lớn nhất” trong mảng một chiều các số thực. Nếu mảng không có giá trị âm thì trả về giá trị 0.
18. Tìm “số nguyên tố lớn nhất” trong mảng một chiều các số nguyên. Nếu mảng không có số nguyên tố thì trả về giá trị 0.
19. Tìm “số hoàn thiện nhỏ nhất” trong mảng một chiều các số nguyên. Nếu mảng không có số hoàn thiện thì trả về giá trị 0.
20. Tìm “giá trị chẵn nhỏ nhất” trong mảng một chiều các số nguyên. Nếu mảng không có giá trị chẵn thì trả về giá trị không chẵn là -1.
21. Tìm “vị trí giá trị âm lớn nhất” trong mảng các số nguyên thực. Nếu mảng không có giá trị âm thì trả về -1.
22. Viết chương trình kiểm tra các phần tử trong một mảng có tăng liên tục hay không.
23. Viết chương trình tìm UCLN của một mảng n phần tử nguyên dương.
24. Viết chương trình xuất ra n phần tử đầu tiên của dãy Fibonacci.
25. Viết chương trình tìm số lớn nhất và nhỏ nhất trong mảng số nguyên A cho trước với số phần tử của mảng là n .
26. Viết chương trình tìm và đổi chỗ phần tử lớn nhất và phần tử nhỏ nhất trong mảng.
27. Cho mảng một chiều các số nguyên. Hãy viết hàm tìm giá trị đầu tiên trong mảng thỏa tính chất số gánh. (Ví dụ giá trị 12321)
28. Tìm giá trị đầu tiên trong mảng một chiều các số nguyên có chữ số đầu tiên là chữ số lẻ (dauledautien). Nếu trong mảng không tồn tại giá trị như vậy hàm sẽ trả về giá trị 0.
29. Cho mảng một chiều các số thực hãy tìm đoạn $[a, b]$ sao cho đoạn này chứa tất cả các giá trị trong mảng
30. Cho mảng một chiều các số thực và một số thực x . Hãy viết hàm tìm giá trị âm cuối cùng lớn hơn giá trị x . Nếu mảng không có giá trị thỏa thì hàm trả về giá trị không chẵn là 0.

1. Nhập vào một mảng có n số nguyên. Xuất ra mảng vừa nhập

```
#include<stdio.h>
#define MAX 999

void nhap (int &n, int mang[] );
void xuat(int n, int mang[]);
```

```

int main()
{
    int n;
    int mang[MAX];
    nhap(n,mang);
    xuat(n,mang);
    return 0;
}
void nhap (int &n, int mang[])
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&mang[i]);
    }
}
void xuat(int n, int mang[])
{
    for(int i=0;i<n;i++)
    {
        printf(" mang[%d]= %d  ",i,mang[i]);
    }
}

```

2. Nhập vào mảng có n số thực. Tính tổng các phần tử trong mảng.

```

#include<stdio.h>
#define MAX 999

void nhap (int &n, float mang[]);
float tongmang(int n, float mang[]);
void xuat(float kq);

int main()
{
    int n;
    float mang[MAX];
    nhap(n,mang);
    float kq=tongmang(n,mang);
    xuat(kq);
    return 0;
}
void nhap (int &n, float mang[])
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%f",&mang[i]);
    }
}
float tongmang(int n, float mang[])
{
    float t=0;
    for(int i=0;i<n;i++)
    {
        t=t+mang[i];
    }
    return t;
}

```

```

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

```

3. Viết chương trình tìm số lớn nhất trong một mảng số nguyên

```

// tìm số lớn nhất trong mảng

#include <iostream>
#define max 500
using namespace std;

void nhap(int A[], int &n);
int solonnhatmang(int A[], int n);
void xuat(int kq);

int main()
{
    int n, A[max];
    nhap(A,n);
    int kq=solonnhatmang(A,n);
    xuat(kq);
    return 0;
}

void nhap(int A[],int &n)
{
    cin>>n;
    for (int i=0; i<n; i++)
        cin>>A[i];
}

int solonnhatmang(int A[], int n)
{
    int m=A[0];
    for (int i=1; i<n; i++)
    {
        if (A[i]>m)
            m=A[i];
    }
    return m;
}

void xuat(int kq)
{
    cout<<kq;
}

```

4. Viết chương trình tính trung bình cộng của một mảng n số nguyên

```

// tìm số lớn nhất trong mảng

#include <iostream>
#define max 999
using namespace std;

void nhap(int A[], int &n);
float tbc(int A[], int n);

```

```

void xuat(float kq);

int main()
{
    int n, A[max];
    nhap(A,n);
    float kq=tbc(A,n);
    xuat(kq);
    return 0;
}

void nhap(int A[],int &n)
{
    cin>>n;
    for (int i=0; i<n; i++)
        cin>>A[i];
}

float tbc(int A[], int n)
{
    int m=0;
    for (int i=0; i<n; i++)
    {
        m+=A[i];
    }
    float e=(float)m/n;
    return e;
}

void xuat(float kq)
{
    cout<<kq;
}

```

5. Viết chương trình tính trung bình cộng của các số chia hết cho 5 trong mảng.

```

// tìm số lớn nhất trong mảng

#include <stdio.h>
#define max 500

void nhap(float A[], int& n);
float tbc(float A[], int n);
void xuat(float kq);

int main()
{
    int n;
    float A[max];
    nhap(A, n);
    float kq = tbc(A, n);
    xuat(kq);
    return 0;
}

void nhap(float A[], int& n)

```

```

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

float tbc(float A[], int n)
{
    int m = 0;
    int a = 0;
    for (int i = 0; i < n; i++)
    {
        if ((int)A[i] % 5 == 0)
        {
            a++;
            m += A[i];
        }
    }
    float e = (float)m / a;
    return e;
}

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

```

6. Viết chương trình đếm số nguyên tố có trong một mảng các số nguyên.

```

#include<stdio.h>
#define MAX 999

void nhap (int &n, int mang[]);
int nguyento( int x );
int demsonguyentotrongmang(int n, int mang[]);
void xuat(int kq);
int main()
{
    int n;
    int mang[MAX];
    nhap(n,mang);
    int kq=demsonguyentotrongmang(n,mang);
    xuat(kq);
}
void nhap (int &n, int mang[])
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&mang[i]);
    }
}
int nguyento( int x )
{

```

```

    int d = 0;
    for ( int i = 1 ; i<= x ; i++)
    {
        if( x % i == 0)
            d ++;          //    dem = dem + 1;
    }
    if(d==2)
    return 1;
    return 0;
}
int demsonguyentotrongmang(int n, int mang[])
{
    int dem=0;
    for(int i=0;i<n;i++)
    {
        if(nguyento(mang[i])==1)
            dem++;
    }
    return dem;
}

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

```

7. Viết chương trình đếm số lượng số hoàn thiện trong một mảng.

```

#include<stdio.h>
#define MAX 999

void nhap (int &n, int mang[]);
int sohoanhao (int a);
int demsohoanthientrongmang(int n, int mang[]);
void xuat(int kq);
int main()
{
    int n;
    int mang[MAX];
    nhap(n,mang);
    int kq=demsohoanthientrongmang(n,mang);
    xuat(kq);
}
void nhap (int &n, int mang[])
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&mang[i]);
    }
}
int sohoanhao (int x)
{
    int s=0;
    for (int i=0; i<x ;i++) // khong lay so chia het cho chinh no la no
    nen i<a ma khong i<=a
    {

```

```

        if (x % i==0)
            s=s+i;
    }
    if (s==x)
        return 1;
    return 0;
}

int demsohoanthientrongmang(int n, int mang[])
{
    int dem=0;
    for(int i=0;i<n;i++)
    {
        if(sohoanhao(mang[i])==1)
            dem++;
    }
    return dem;
}

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

```

8. Viết chương trình tính trung bình cộng của các số nguyên tố trong mảng.

```

#include<stdio.h>
#define MAX 999

void nhap (int &n, int mang[]);
int nguyento( int x );
int tbcsonguyentotrongmang(int n, int mang[]);
void xuat(int kq);
int main()
{
    int n;
    int mang[MAX];
    nhap(n,mang);
    int kq=tbcsonguyentotrongmang(n,mang);
    xuat(kq);
}

void nhap (int &n, int mang[])
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&mang[i]);
    }
}

int nguyento( int x )
{
    int d = 0;
    for ( int i = 1 ; i<= x ; i++)
    {
        if( x % i == 0)

```

```

        d ++;        //    dem = dem + 1;
    }
    if(d==2)
    return 1;
    return 0;
}
int tbcsongyuentotrongmang(int n, int mang[])
{
    int dem=0;
    for(int i=0;i<n;i++)
    {
        if(nguyento(mang[i])==1)
            dem+=mang[i];
    }
    return dem;
}

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

```

9. Tìm giá trị âm đầu tiên trong mảng một chiều các số âm thực. Nếu mảng không có giá trị âm thì trả về giá trị không âm là giá trị 1.

```

#include<stdio.h>
#define MAX 999

void nhap (int &n, float mang[]);
float timsoamdautientrongmang(int n,float mang[]);
void xuat(float kq);
int main()
{
    int n;
    float mang[MAX];
    nhap(n,mang);
    float kq=timsoamdautientrongmang(n,mang);
    xuat(kq);
    return 0;
}

void nhap (int &n, float mang[])
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%f",&mang[i]);
    }
}

float timsoamdautientrongmang(int n,float mang[])
{
    for(int i=0;i<n;i++)
    {
        if(mang[i]<0)
            return mang[i];
    }
    return 1;
}

```



```

void xuat(float kq)
{
    printf("%.0f",kq);
}

```

10. Viết hàm tìm “số chẵn đầu tiên” trong mảng các số nguyên. Nếu mảng không có giá trị chẵn thì hàm sẽ trả về giá trị không chẵn là -1.

```

#include<stdio.h>
#define MAX 999

void nhap (int &n, int mang[]);
float timsochandautiontrongmang(int n,int mang[]);
void xuat(int kq);
int main()
{
    int n;
    int mang[MAX];
    nhap(n,mang);
    int kq=timsochandautiontrongmang(n,mang);
    xuat(kq);
    return 0;
}
void nhap (int &n, int mang[])
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&mang[i]);
    }
}
float timsochandautiontrongmang(int n,int mang[])
{
    for(int i=0;i<n;i++)
    {
        if(mang[i]%2==0)
            return mang[i];
    }
    return -1;
}
void xuat(int kq)
{
    printf("%d",kq);
}

```

11. Tìm “số nguyên tố đầu tiên” trong mảng một chiều các số nguyên. Nếu mảng không có số nguyên tố thì trả về giá trị -1.

```

#include<stdio.h>
#define MAX 999

void nhap (int &n, int mang[]);
float timsonguyentodautientrongmang(int n,int mang[]);
void xuat(int kq);
int main()
{
    int n;

```

```

        int mang[MAX];
        nhap(n,mang);
        int kq=timsonguyentodautientrongmang(n,mang);
        xuất(kq);
        return 0;
    }
void nhap (int &n, int mang[])
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&mang[i]);
    }
}
int nguyento( int x )
{
    int d = 0;
    for ( int i = 1 ; i<= x ; i++)
    {
        if( x % i == 0 )
            d ++;          //    dem = dem + 1;
    }
    if(d==2)
        return 1;
    return 0;
}
float timsonguyentodautientrongmang(int n,int mang[])
{
    for(int i=0;i<n;i++)
    {
        if(nguyento(mang[i])==1)
            return mang[i];
    }
    return -1;
}
void xuất(int kq)
{
    printf("%d",kq);
}
}

```

12. Tìm “số hoàn thiện đầu tiên” trong mảng một chiều các số nguyên. Nếu mảng không có số hoàn thiện thì trả về giá trị -1.

```

#include<stdio.h>
#define MAX 999

void nhap (int &n, int mang[]);
int sohoanhao (int x);
int timsohoanthientrongmang(int n, int mang[]);
void xuất(int kq);
int main()
{
    int n;
    int mang[MAX];
    nhap(n,mang);
    int kq=timsohoanthientrongmang(n,mang);
}

```

```

        xuất(kq);
    }
    void nhap (int &n, int mang[])
    {
        scanf("%d",&n);
        for(int i=0;i<n;i++)
        {
            scanf("%d",&mang[i]);
        }
    }
    int sohoanhao (int x)
    {
        int s=0;
        for (int i=0; i<x ;i++) // không lấy số chia hết cho chính nó là nó
        nen i<a ma không i<=a
        {
            if (x % i==0)
                s=s+i;
        }
        if (s==x)
            return 1;
        return 0;
    }

    int timsohoanthientrongmang(int n, int mang[])
    {
        for(int i=0;i<n;i++)
        {
            if(sohoanhao(mang[i])==1)
                return mang[i];
        }
        return -1;
    }

    void xuất(int kq)
    {
        printf("%d", kq);
    }
}

```

13. Tìm “số nguyên dương cuối cùng” trong mảng số thực. Nếu mảng không có giá trị dương thì trả về giá trị -1.

```

#include<stdio.h>
#define MAX 999
void nhap (int &n, int mang[]);
int songuyencuoicung(int n, int mang[]);
void xuất(int kq);

int main()
{
    int n;
    int mang[MAX];
    nhap(n,mang);
    int kq=songuyencuoicung(n,mang);
    xuất(kq);
}

```

```

}
void nhap (int &n, int mang[])
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&mang[i]);
    }
}

int songuyencuoicung(int n, int mang[])
{
    for(int i=n-1;i>=0;i--)
    {
        if((mang[i]%2)==0)
            return mang[i];
    }
    return -1;
}

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

```

14. Tìm “số nguyên tố cuối cùng” trong mảng một chiều các số nguyên. Nếu mảng không có số nguyên tố thì trả về giá trị -1.

```

#include<stdio.h>
#define MAX 999
void nhap (int &n, int mang[]);
int nguyento( int x );
int songuyentocuoicung(int n, int mang[]);
void xuat(int kq);

int main()
{
    int n;
    int mang[MAX];
    nhap(n,mang);
    int kq=songuyentocuoicung(n,mang);
    xuat(kq);
}

void nhap (int &n, int mang[])
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&mang[i]);
    }
}

int nguyento( int x )
{
    int dem = 0;
    for ( int i = 1 ; i <= x ; i++)
    {

```

```

        if( x % i == 0)
            dem ++;          //    dem = dem + 1;
    }
    if(dem==2)
    return 1;
    return 0;
}

int songuyentocuoicung(int n, int mang[])
{
    for(int i=n-1;i>=0;i--)
    {
        if(nguyento(mang[i]))
            return mang[i];
    }
    return -1;
}

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

```

15. Tìm “số hoàn thiện cuối cùng” trong mảng một chiều các số nguyên. Nếu mảng không có số hoàn thiện thì trả về giá trị 0.

```

#include<stdio.h>
#define MAX 999
void nhap (int &n, int mang[]);
int nguyento( int x );
int sohoanhaocuoicung(int n, int mang[]);
void xuat(int kq);

int main()
{
    int n;
    int mang[MAX];
    nhap(n,mang);
    int kq=sohoanhaocuoicung(n,mang);
    xuat(kq);
}

void nhap (int &n, int mang[])
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&mang[i]);
    }
}

int sohoanhao (int x)
{
    int s=0;
    for (int i=0; i<x ;i++) // khong lay so chia het cho chinh no la no
    nen i<a ma khong i<=a
    {
        if (x % i==0)

```

```

        s=s+i;
    }
    if (s==x)
        return 1;
    return 0;
}

int sohoanhaocuoicung(int n, int mang[])
{
    for(int i=n-1;i>=0;i--)
    {
        if(sohoanhao(mang[i]))
            return mang[i];
    }
    return 0;
}

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

```

16. Viết chương trình tìm số âm lớn nhất trong một mảng các số nguyên.

```

#include<stdio.h>
#define MAX 999
void nhap(int &n,int A[]);
int tinsoamlonnhhat(int n, int A[]);
void xuat (int kq);

int main()
{
    int n,A[MAX];
    nhap(n,A);
    int kq=tinsoamlonnhhat(n,A);
    xuat(kq);
}

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

int tinsoamlonnhhat(int n, int A[])
{
    int max;
    max=A[0];
    for(int i=0; i<n; i++)
    {
        if(A[i]<0)
        {
            if(A[i]>max)
                max=A[i];
        }
    }
}

```

```

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

```

17. Tìm số “âm lớn nhất” trong mảng một chiều các số thực. Nếu mảng không có giá trị âm thì trả về giá trị 0.

```

#include<stdio.h>
#define MAX 999
void nhap(int &n,int A[]);
int tinsoamlonnhat(int n, int A[]);
void xuat (int kq);

int main()
{
    int n,A[MAX];
    nhap(n,A);
    int kq=tinsoamlonnhat(n,A);
    xuat(kq);
}
void nhap (int &n, int A[])
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&A[i]);
    }
}
int tinsoamlonnhat(int n, int A[])
{
    int max;
    max=A[0];
    for(int i=0; i<n; i++)
    {
        if(A[i]<0)
        {
            if(A[i]>max)
                max=A[i];
        }
    }
    if(max<0)
    return max;
    return 0;
}
void xuat (int kq)
{
    printf("%d",kq);
}

```

18. Tìm “số nguyên tố lớn nhất” trong mảng một chiều các số nguyên. Nếu mảng không có số nguyên tố thì trả về giá trị 0.

```
#include<stdio.h>
#define MAX 999
void nhap (int &n, int A[]);
int timsonguyentolonnhhat(int n, int A[], int B[]);
void xuat (int kq);

int main()
{
    int n,A[MAX],B[MAX];
    nhap(n,A);
    int kq=timsonguyentolonnhhat(n,A,B);
    xuat(kq);
}
void nhap (int &n, int A[])
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&A[i]);
    }
}
int nguyento( int x )
{
    int dem = 0;
    for ( int i = 1 ; i <= x ; i++)
    {
        if( x % i == 0 )
            dem ++;           //    dem = dem + 1;
    }
    if(dem==2)
        return 1;
    return 0;
}

int timsonguyentolonnhhat(int n, int A[], int B[])
{
    int gan;
    int dem=0;
    for(int i=0; i<n; i++)
    {
        if(nguyento(A[i])==1)
        {
            gan=A[i];
            B[i]=gan;
            dem++;
        }
    }
    if(dem>0)
    {
```



```

        int max;
        max=B[0];
        for(int j=0; j<n; j++)
        {
            if(B[j]>max)
                max=B[j];
        }
        return max;
    }
    return 0;
}
void xuat (int kq)
{
    printf("%d",kq);
}
}

```

19. Tìm “số hoàn thiện nhỏ nhất” trong mảng một chiều các số nguyên. Nếu mảng không có số hoàn thiện thì trả về giá trị 0.

```

#include<stdio.h>
#define MAX 999
void nhap (int &n, int A[]);
int sohoanhao (int a);
int timsohoanhaonhonhat(int n, int A[], int B[]);
void xuat (int kq);
int main()
{
    int n,A[MAX],B[MAX];
    nhap(n,A);
    int kq=timsohoanhaonhonhat(n,A,B);
    xuat(kq);
}
void nhap (int &n, int A[])
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&A[i]);
    }
}
int sohoanhao (int a)
{
    int s=0;
    for (int i=0; i<a ;i++) // không lấy số chia hết cho chính nó là nó
    nen i<a ma không i<=a
    {
        if (a % i==0)
            s=s+i;
    }
    if (s==a)
        return 1;
    else
        return 0;
}
}

```

```

int timsohoanhaonhonhat(int n, int A[], int B[])
{
    int gan;
    int dem=0;
    for(int i=0; i<n; i++)
    {
        if(sohoanhao(A[i])==1)
            gan=A[i];
        B[i]=gan;
        dem++;
    }
    if(dem>0)
    {
        int min;
        min=B[0];
        for(int j=0; j<n; j++)
        {
            if(B[j]<min)
                min=B[j];
        }
        return min;
    }
    return 0;
}

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

```

20. Tìm “giá trị chẵn nhỏ nhất” trong mảng một chiều các số nguyên. Nếu mảng không có giá trị chẵn thì trả về giá trị không chẵn là -1.

```

#include<stdio.h>
#define MAX 999
void nhap (int &n, int A[]);
int giatrichannhonhat(int n, int A[], int B[]);
void xuat (int kq);

int main()
{
    int n,A[MAX],B[MAX];
    nhap(n,A);
    int kq=giatrichannhonhat(n,A,B);
    xuat(kq);
}

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

```

```

    }
}

int giatrichannhonhat(int n, int A[], int B[])
{
    int dem=0;
    int gan;
    for(int i=0; i<n; i++)
    {
        if(A[i]%2==0)
        {
            dem++;
            gan=A[i];
            B[i]=gan;
        }
    }
    if(dem>0)
    {
        int min;
        min=B[0];
        for(int j=0; j<n; j++)
        {
            if(B[j]<min)
                min=B[j];
        }
        return min;
    }
    return -1;
}

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

```

21. Tìm “vị trí giá trị âm lớn nhất” trong mảng các số nguyên thực. Nếu mảng không có giá trị âm thì trả về -1.

```

#include<stdio.h>
#define MAX 999
void nhap (int &n, int A[]);
int vitrigiatriamax(int n, int A[], int B[]);
void xuat (int kq);

int main()
{
    int n,A[MAX],B[MAX];
    nhap(n,A);
    int kq=vitrigiatriamax(n,A,B);
    xuat(kq);
}

void nhap (int &n, int A[])
{
    scanf("%d",&n);
}

```

```

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

int vitrigiatriamax(int n, int A[], int B[])
{
    int dem=0;
    int gan;
    for(int i=0; i<n; i++)
    {
        if(A[i]<0)
        {
            dem++;
            gan=A[i];
            B[i]=gan;
        }
    }
    if(dem>0)
    {
        int max;
        max=B[0];
        int vt;
        for(int j=0; j<n; j++)
        {
            if(B[j]>max)
            {
                max=B[j];
                vt=j+1;
            }
        }
        return vt;
    }
    return -1;
}

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

```

22. Viết chương trình kiểm tra các phần tử trong một mảng có tăng liên tục hay không.

```

#include<stdio.h>
#define MAX 999
void nhap (int &n, int A[]);
int cacphantutanglientuc(int n, int A[]);
void xuat (int kq);
int main()
{
    int n,A[MAX];
    nhap(n,A);
    int kq=cacphantutanglientuc(n,A);
    xuat(kq);
}

```

```

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

int cacphantutanglientuc(int n, int A[])
{
    int i=1;
    while(i<n, A[i-1]<A[i])
        i++;
    if(i==n)
        return 1;
    return 0;
}

void xuat (int kq)
{
    if(kq==1)
        printf("tang lien tuc");
    else
        printf("khong tang lien tuc");
}

```

23. Viết chương trình tìm UCLN của một mảng n phần tử nguyên dương.

```

#include<stdio.h>
#define MAX 999
void nhap (int &n, int A[]);
int usclnhaiso(int a, int b);
int UCLNtrongmang(int n, int A[]);
void xuat (int kq);

int main()
{
    int n,A[MAX];
    nhap(n,A);
    int kq=UCLNtrongmang(n,A);
    xuat(kq);
}

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

```

```

int usclnhaiso(int a, int b)
{
    int c;
    if (a < b)
    {
        for (int i = a; i >= 1; i--)
        {
            if (a % i == 0 && b % i == 0)
                c = i;
            break;
        }
    }
    if (b < a)
    {
        for (int i = b; i >= 1; i--)
        {
            if (a % i == 0 && b % i == 0)
                c = i;
            break;
        }
    }
    return c;
}

int UCLNtrongmang(int n, int A[])
{
    int UCLN = A[0];
    for (int i = 1; i < n; i++)
    {
        UCLN = usclnhaiso(UCLN, A[i]);
    }
    return UCLN;
}

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

```

24. Viết chương trình xuất ra n phần tử đầu tiên của dãy Fibonacci.

```

#include<stdio.h>
#define MAX 999
void nhap (int &n);
void fibonacci(int n, int B[]);
void xuatra(int B[], int n);
void xuat (int i);

int main()
{
    int n,B[MAX];
    nhap(n);
    fibonacci(n,B);
}

```

```

}

void nhap (int &n)
{
    scanf("%d",&n);
}
void fibonacci(int n, int B[])
{
    int gan;
    int n3;
    int n1=0;
    int n2=1;
    for(int i=2;i<n;i++)    //vong lap bat dau tu 2 boi vi 0 va 1 da duoc
in
        {
            n3=n1+n2;
            gan=n3;
            B[i-1]=n3;
            n1=n2;
            n2=n3;
        }

}
void xuatra(int B[], int n)
{
    for(int i=0; i<n; i++)
    {
        xuat(B[i]);
    }
}

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

```

25. Viết chương trình tìm số lớn nhất và nhỏ nhất trong mảng số nguyên A cho trước với số phần tử của mảng là n.

```

#include<stdio.h>
#define MAX 999
void nhap (int &n, int A[]);
int solonnhat(int n, int A[]);
int sonhonhat(int n, int A[]);
void xuat (int kq, int kq1);

int main()
{
    int n,A[MAX];
    nhap(n,A);
    int kq=solonnhat(n,A);
    int kq1=sonhonhat(n,A);
    xuat(kq,kq1);
}

```

```

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

int solonnhat(int n, int A[])
{
int max = A[0];
    for (int i = 1; i < n; i++)
    {
        if(A[i]>max)
            max=A[i];
    }
    return max;
}

int sonhonhat(int n, int A[])
{
    int min = A[0];
    for (int i = 1; i < n; i++)
    {
        if(A[i]<min)
            min=A[i];
    }
    return min;
}

void xuat (int kq, int kq1)
{
    printf("%d %d ",kq,kq1);
}

```

26. Viết chương trình tìm và đổi chỗ phần tử lớn nhất và phần tử nhỏ nhất trong mảng.

```

#include<stdio.h>
void hoandoi(float A[], int vtmin, int vtmax);
void Xuat2(float a);
void MaxMin(float A[], int n);
void Xuat1(float min, float max);
void Nhap(float A[], int& n);

int main()
{
    int n;
    float A[100];
    Nhap(A, n);
    MaxMin(A, n);
}

```



```

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

void hoandoi(float A[], int vtmin, int vtmax)
{
    float d;
    d = A[vtmin];
    A[vtmin] = A[vtmax];
    A[vtmax] = d;
}

void MaxMin(float A[], int n)
{
    int vtmin = 0, vtmax = 0;
    float min = A[0];
    float max = A[0];
    for (int i = 1; i < n; i++)
    {
        if (min > A[i])
        {
            min = A[i];
            vtmin = i;
        }
        else if (max < A[i])
        {
            max = A[i];
            vtmax = i;
        }
    }
    Xuat1(min, max);
    hoandoi(A, vtmin, vtmax);
    for (int i = 0; i < n; i++)
        Xuat2(A[i]);
}

void Xuat1(float min, float max)
{
    printf("Phan tu nho nhat: %.2f\nPhan tu lon nhat: %.2f\n", min, max);
}

void Xuat2(float a)
{
    printf("%.2f ", a);
}

```

27 Cho mảng một chiều các số nguyên. Hãy viết hàm tìm giá trị đầu tiên trong mảng thỏa tính chất số gánh. (Ví dụ giá trị 12321)

```

#include<stdio.h>
#define MAX 999
void nhap (int &n, int A[]);
int nghichdao(int a);
int timsoganhdautien(int n, int A[]);
void xuat(int kq);

```

```

int main()
{
    int n,A[MAX];
    nhap(n,A);
    int kq=timsoganhdautien(n,A);
    xuat(kq);
}
void nhap (int &n, int A[])
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%d",&A[i]);
    }
}
int nghichdao(int a)
{
    int ketqua = 0;
    while (a > 0)
    {
        int chusocuoi = a % 10;
        a= a / 10;
        ketqua =ketqua*10+chusocuoi;
    }
    if(ketqua==a)
    return 1;
}
int timsoganhdautien(int n, int A[])
{
    for(int i=0; i<n; i++)
    {
        if(nghichdao(A[i]))
        return A[i];
    }
}
void xuat(int kq)
{
    printf("%d",kq);
}

```

28. Tìm giá trị đầu tiên trong mảng một chiều các số nguyên có chữ số đầu tiên là chữ số lẻ (dauledautien). Nếu trong mảng không tồn tại giá trị như vậy hàm sẽ trả về giá trị 0.

```

#include<stdio.h>
#define MAX 999
void nhap (int &n, int A[]);
int timsoledautien(int n, int A[]);
void xuat(int kq);

int main()
{
    int n,A[MAX];
    nhap(n,A);
    int kq=timsoledautien(n,A);
    xuat(kq);
}

```

```

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

int timsoledautien(int n, int A[])
{
    for(int i=0; i<n; i++)
    {
        if(A[i]%2!=0)
            return A[i];
    }
    return 0;
}

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

```

29. Cho mảng một chiều các số thực hãy tìm đoạn $[a, b]$ sao cho đoạn này chứa tất cả các giá trị trong mảng

```

#include<stdio.h>
#define MAX 999
void nhap (float &n, float A[]);
void maxmin(float n, float A[]);
void xuat(float min, float max);

int main()
{
    float n,A[MAX];
    nhap(n,A);
    maxmin(n,A);
}

void nhap (float &n, float A[])
{
    scanf("%f",&n);
    for(int i=0;i<n;i++)
    {
        scanf("%f",&A[i]);
    }
}

void maxmin(float n, float A[])
{
    int vtmin = 0, vtmax = 0;
    float min = A[0];
    float max = A[0];
    for (int i = 1; i < n; i++)
    {
        if (min > A[i])
        {
            min = A[i];

```

```

        vtmin = i;
    }
    else if (max < A[i])
    {
        max = A[i];
        vtmax = i;
    }
}
xuat(min, max);

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

```

30. Cho mảng một chiều các số thực và một số thực x. Hãy viết hàm tìm giá trị âm cuối cùng lớn hơn giá trị x. Nếu mảng không có giá trị thỏa thì hàm trả về giá trị không chẵn là 0.

```

#include<stdio.h>
#define MAX 999
void nhap (float &n,float &x, float A[]);
void soamlonnhat(float n,int x, float A[]);
void xuat(float kq);

int main()
{
    float x,n,A[MAX];
    nhap(x,n,A);
    float kq=soamlonnhat(x,n,A);
    xuat(kq);
}
void nhap (float &x,float &n, float A[])
{
    scanf("%f%f",&x,&n);
    for(int i=0;i<n;i++)
    {
        scanf("%f",&A[i]);
    }
}
void soamlonnhat(float n,int x, float A[])
{
    for (int i = n-1; i >=0; i--)
    {
        min=A[n-1];
        if (A[i]<0&&A[i]<min;]&&A[i]>x)
        {
            return A[i];
        }
    }
    return 0;
}
void xuat(float kq)
{
    printf("%.2f",kq);
}

```

NHÓM BÀI TẬP SỐ 3

1	2
<pre> #include<stdio.h> void nhapMang(int A[], int &n); void xuatMang(int A[], int n); void main() { int A[100],n; nhapMang(A,n); xuatMang(A,n); } 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]); } </pre>	<pre> #include<stdio.h> void nhapMang(float A[], int &n); void xuatMang(float A[], int n); float tinhTong(float A[], int n); void xuat(float Tong); void main() { int n; float A[100]; nhapMang(A,n); xuatMang(A,n); float kq=tinhTong(A,n); xuat(kq); } void nhapMang(float A[], int &n) { scanf("%d",&n); for(int i=0;i<n;i++) scanf("%f",&A[i]); } void xuatMang(float A[], int n) { for(int i=0;i<n;i++) printf("%.2f ",A[i]); } float tinhTong(float A[], int n) { float Tong=0; for(int i=0;i<n;i++) { Tong=Tong+A[i]; } return Tong; } void xuat(float Tong) { printf("Tong: %.2f\n",Tong); } </pre>
<pre> #include<stdio.h> void nhapMang(int A[], int &n); int tinhTong(int A[], int n); void xuat(int max); void main() { int A[100],n; nhapMang(A,n); int kq=tinhTong(A,n); xuat(kq); } void nhapMang(int A[], int &n) { scanf("%d",&n); for (int i=0;i<n;i++) scanf("%d",&A[i]); } int tinhTong(int A[], int n) { int max=0; for(int i=0;i<n;i++) { if(max<A[i]) { max=A[i]; } } return max; } void xuat(int max) { printf("%d",max); } </pre>	<pre> #include<stdio.h> void nhapMang(int A[], int &n); float tinhTb(int A[], int n); void xuat(float Ttb); void main() { int A[200],n; nhapMang(A,n); float kq=tinhTb(A,n); xuat(kq); } void nhapMang(int A[], int &n) { scanf("%d",&n); for(int i=0;i<n;i++) scanf("%d",&A[i]); } </pre>
3	4
5	5

```

#include<stdio.h>
void nhapMang( int A[], int &n);
float tinhTbsochiahetcho5( int A[], int n);
void xuat(float tb);
void main()
{
    int A[120],n;
    nhapMang(A,n);
    float kq=tinhTbsochiahetcho5(A,n);
    xuat(kq);
}
void nhapMang( int A[], int &n)
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%d",&A[i]);
}
float tinhTbsochiahetcho5( int A[], int n)
{
    int i,s,d;
    float tb;
    s=0,d=0;
    for(i=0;i<n;i++)
    {
        if(A[i]%5==0)
        {
            s=s+A[i];
            d=d+1;
        }
    }
    tb=s/d;
    return tb;
}
void xuat(float tb)
{
    printf("%.2f",tb);
}

```

7

```

#include<stdio.h>
void nhapMang( int A[], int &n);
int dem(int A[], int n);
void xuat(int dem);
void main()
{
    int A[150],n;
    nhapMang(A,n);
    int kq=dem(A,n);
    xuat(kq);
}
void nhapMang( int A[], int &n)
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%d",&A[i]);
}
int dem(int A[], int n)
{
    int dem;

```

```

}
float tinhTb(int A[], int n)
{
    float Ttb,s;
    s=0;
    for(int i=0;i<n;i++)
    {
        s=s+A[i];
    }
    Ttb=s/n;
    return Ttb;
}
void xuat(float Ttb)
{
    printf("%.2f",Ttb);
}

```

6

```

#include<stdio.h>
void nhapMang(int A[], int &n);
int dem( int A[], int n);
void xuat(int dem);
void main()
{
    int A[130],n;
    nhapMang(A,n);
    int kq=dem(A,n);
    xuat(kq);
}
void nhapMang(int A[], int &n)
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%d",&A[i]);
}
int dem( int A[], int n)
{
    int dem;
    dem=0;
    for(int i=0;i<n;i++)
    {
        int dem1=0;
        for(int j=1;j<=A[i];j++)
        {
            if(A[i]%j==0)
            {
                dem1=dem1+1;
            }
        }
        if(dem1==2)
        {
            dem=dem+1;
        }
    }
    return dem;
}
void xuat(int dem)
{

```

```

dem=0;
for(int i=0; i<n;i++)
{
    int s=0;
    for(int j=1;j<A[i];j++)
    {
        if(A[i]%j==0)
        {
            s=s+j;
        }
    }
    if(s==A[i])
    {
        dem=dem+1;
    }
}
return dem;
}
void xuat(int dem)
{
    printf("%d",dem);
}

```

9

```

#include<stdio.h>
void nhapMang(float A[], int &n);
float kiemtra(float A[], int n);
void main()
{
    float A[200];
    int n;
    nhapMang(A,n);
    kiemtra(A,n);
    printf("%.2f",kiemtra(A,n));
}
void nhapMang(float A[], int &n)
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%f",&A[i]);
}
float kiemtra(float A[], int n)
{
    for(int i=0;i<n;i++)
    {
        if(A[i]<0)
            return A[i];
    }
    return 1;
}

```

11

```

#include<stdio.h>
void nhapMang(int A[], int &n);
int hienra(int A[],int n);
void main()
{

```

```

printf("%d",dem);
}
8
#include<stdio.h>
void nhapMang( int A[], int &n);
float tinhthb( int A[], int n);
void xuat(float tb);
void main()
{
    int A[210],n;
    nhapMang(A,n);
    float kq=tinhthb(A,n);
    xuat(kq);
}
void nhapMang( int A[], int &n)
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%d",&A[i]);
}
float tinhthb( int A[], int n)
{

```

```

float tb,s,dem;
dem=0;
s=0;
for(int i=0;i<n;i++)
{
    int j,dem1;
    dem1=0;
    j=1;
    while(j<=A[i])
    {
        if(A[i]%j==0)
        {
            dem1=dem1+1;
        }
        j++;
    }
    if(dem1==2)
    {
        s=s+A[i];
        dem=dem+1;
    }
}
tb=s/dem;
return tb;
}
void xuat(float tb)
{
    printf("%.2f",tb);
}

```

10

```

#include<stdio.h>
void nhapMang(int A[], int &n);
int kiemtra(int A[], int n);
void main()
{

```

```

int A[150],n;
nhapMang(A,n);
hienra(A,n);
printf("%d",hienra(A,n));
}
void nhapMang(int A[], int &n)
{
scanf("%d",&n);
for(int i=0;i<n;i++)
scanf("%d",&A[i]);
}
int hienra(int A[],int n)
{
int dem;
for(int i=0;i<n;i++)
{
dem=0;
for(int j=1;j<=A[i];j++)
{
if(A[i]%j==0)
{
dem=dem+1;
}
}
if(dem==2)
{
return A[i];
}
}
if(dem!=2)
{
return -1;
}
}
13
#include<stdio.h>
void nhapMang(float A[], int &n);
float hien( float A[], int n);
void main()
{
float A[150];
int n;
nhapMang(A,n);
hien(A,n);
printf("%.2f",hien(A,n));
}
void nhapMang(float A[], int &n)
{
scanf("%d",&n);
for(int i=0;i<n;i++)
scanf("%f",&A[i]);
}
float hien( float A[], int n)
{
for(int i=n-1;i>=0;i--)
{
int x;
x=int(A[i]);

```

```

int A[150],n;
nhapMang(A,n);
kiemtra(A,n);
printf("%d",kiemtra(A,n));
}
void nhapMang(int A[], int &n)
{
scanf("%d",&n);
for(int i=0;i<n;i++)
scanf("%d",&A[i]);
}
int kiemtra(int A[], int n)
{
for(int i=0;i<n;i++)
{
if(A[i]%2==0)
{
return A[i];
}
}
return -1;
}
12
#include<stdio.h>
void nhapMang(int A[], int &n);
int hien(int A[], int n);
void main()
{
int A[100],n;
nhapMang(A,n);
hien(A,n);
printf("%d",hien(A,n));
}
void nhapMang(int A[], int &n)
{
scanf("%d",&n);
for(int i=0;i<n;i++)
scanf("%d",&A[i]);
}
int hien(int A[], int n)
{
int s;
for(int i=0;i<n;i++)
{
s=0;
for(int j=1;j<A[i];j++)
{
if(A[i]%j==0)
{
s=s+j;
}
}
if(A[i]==s)
{
return s;
}
}
if(A[i]!=s)

```


<pre> if(A[i]>0 && (A[i]-x)==0) return A[i]; } return -1; } 15 #include<stdio.h> void nhapMang(int A[], int &n); int hoanhao(int A[], int n); void xuat(int kq); void main() { int A[150],n; nhapMang(A,n); int kq=hoanhao(A,n); xuat(kq); } void nhapMang(int A[], int &n) { scanf("%d",&n); for(int i=0;i<n;i++) scanf("%d",&A[i]); } int hoanhao(int A[], int n) { for(int i=n-1;i>=0;i--) { int s=0; for(int j=1;j<A[i];j++) { if(A[i]%j==0) { s=s+j; } } if(A[i]==s) { return A[i]; } } return 0; } void xuat(int kq) { printf("%d",kq); } 17 #include<stdio.h> void nhapMang(float A[], int &n); float soam(float A[], int n); void xuat(float kq); void main() { float A[150]; int n; nhapMang(A,n); </pre>	<pre> { return -1; } } 14 #include<stdio.h> void nhapMang(int A[], int &n); int nguyento(int A[], int n); void xuat(int kq); void main() { int A[150],n; nhapMang(A,n); int kq=nguyento(A,n); xuat(kq); } void nhapMang(int A[], int &n) { scanf("%d",&n); for(int i=0;i<n;i++) scanf("%d",&A[i]); } int nguyento(int A[], int n) { for(int i=n-1;i>=0;i--) { int dem=0; for(int j=1;j<=A[i];j++) { if(A[i]%j==0) { dem=dem+1; } } if(dem==2) { return A[i]; } } return -1; } void xuat(int kq) { printf("%d",kq); } 16 #include<stdio.h> void nhapMang(int A[], int &n); int soam(int A[], int n); void xuat(int kq); void main() { int A[150],n; nhapMang(A,n); int kq=soam(A,n); xuat(kq); } void nhapMang(int A[], int &n) </pre>
--	--

<pre> float kq=soam(A,n); xuat(kq); } void nhapMang(float A[], int &n) { scanf("%d",&n); for(int i=0;i<n;i++) scanf("%f",&A[i]); } float soam(float A[], int n) { float max; for(int i=0;i<n;i++) { if(A[i]<0) { max=A[i]; if(A[i]>A[i+1]) { max=A[i]; return max; } } } return 0; } void xuat(float kq) { printf("%.2f",kq); } 19 #include<stdio.h> void nhapMang(int A[], int &n); int tim(int A[], int n); void xuat(int kq); void main() { int A[150],n; nhapMang(A,n); int kq=tim(A,n); xuat(kq); } void nhapMang(int A[], int &n) { scanf("%d",&n); for(int i=0;i<n;i++) scanf("%d",&A[i]); } int tim(int A[], int n) { int min; for(int i=0;i<n;i++) { int s=0; for(int j=1;j<A[i];j++) </pre>	<pre> { scanf("%d",&n); for(int i=0;i<n;i++) scanf("%d",&A[i]); } int soam(int A[], int n) { int max=0; for(int i=0;i<n;i++) { if(A[i]<0) { max=A[i]; if(A[i]>A[i+1]) { max=A[i]; return max; } } } return -1; } void xuat(int kq) { printf("%d",kq); } 18 #include<stdio.h> void nhapMang(int A[], int &n); int Languyento(int x); int timmax(int A[], int n); void xuat(int kq); void main() { int A[150],n; nhapMang(A,n); int kq=timmax(A,n); xuat(kq); } void nhapMang(int A[], int &n) { scanf("%d",&n); for(int i=0;i<n;i++) scanf("%d",&A[i]); } int Languyento(int x) { int dem=0; for(int i=1;i<=x;i++) { if(x%i==0) { dem=dem+1; } } if (dem == 2) </pre>
--	--

<pre> if(A[i]%j==0) { s=s+j; } } if(s==A[i]) { min=A[i]; if(A[i]<A[i+1]) { min=A[i]; return min; } } } return 0; } void xuat(int kq) { printf("%d",kq); } </pre> <p style="text-align: center;">21</p> <pre> #include<stdio.h> void nhapMang(float A[], int &n); int tim (float A[], int n); void xuat(int kq); void main() { float A[150]; int n; nhapMang(A,n); int kq=tim(A,n); xuat(kq); } void nhapMang(float A[], int &n) { scanf("%d",&n); for(int i=0;i<n;i++) scanf("%f",&A[i]); } int tim (float A[], int n) { int maxam; for(int i=0;i<n;i++) { if(A[i]<0) { maxam=A[i]; if(A[i]>A[i+1]) { maxam=A[i]; return i; } } } return -1; } void xuat(int kq) </pre>	<pre> return 1; } else return 0; } } int timmax(int A[], int n) { int max=0; for(int i=0;i<n;i++) { if(Languyento(A[i])==1) { if(A[i] > max) max=A[i]; } } return max; } void xuat(int kq) { printf("%d",kq); } </pre> <p style="text-align: center;">20</p> <pre> #include<stdio.h> void nhapMang(int A[], int &n); int tim(int A[], int n); void xuat(int kq); void main() { int A[150],n; nhapMang(A,n); int kq=tim(A,n); xuat(kq); } void nhapMang(int A[], int &n) { scanf("%d", &n); for(int i=0;i<n;i++) scanf("%d",&A[i]); } int tim(int A[], int n) { int min=0; for(int i=0;i<n;i++) { if(A[i]%2==0) { min=A[i]; if(A[i]<A[i+1]) { min=A[i]; return min; } } } return -1; } </pre>
--	--

```

{
    printf("%d",kq);
}
23
#include<stdio.h>
#include<math.h>
void nhapMang( int A[], int &n);
int UCLN(int A[], int n);
void xuat(int kq);
void main()
{
    int A[100],n;
    nhapMang(A,n);
    int kq=UCLN(A,n);
    xuat(kq);
}
void nhapMang( int A[], int &n)
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%d",&A[i]);
}
int UCLN_haiso( int a ,int b)
{
    while (a!=b)
    {
        if(a>b)
            a=a-b;
        else
            b=b-a;
    }
    return a;
}
int UCLN(int A[], int n)
{
    int temp=abs(A[0]);
    for(int i=0;i<n;i++)
    {
        temp=(UCLN_haiso(temp,abs(A[i])));
    }
    return temp;
}
void xuat(int kq)
{
    printf("%d",kq);
}
25
#include<stdio.h>
void nhapMang( int A[], int &n);
int timMax( int A[], int &n);
int timMin( int A[], int &n);
void xuat(int a, int b);
void main()
{
    int A[150],n;

```

```

void xuat(int kq)
{
    printf("%d",kq);
}
22
#include<stdio.h>
void nhapMang(int A[], int &n);
int tangdeu(int A[], int n );
void xuat(int kq);
void main()
{
    int A[150];
    int n;
    nhapMang(A,n);
    int kq=tangdeu(A,n);
    xuat(kq);
}
void nhapMang(int A[], int &n)
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%d",&A[i]);
}
int tangdeu(int A[], int n )
{
    for(int i=0;i<n-1;i++)
    {
        if(A[i] >= A[i+1])
        {
            return 0;
        }
    }
    return 1;
}
void xuat(int kq)
{
    if( kq==0)
        printf("ko\n");
    else
        printf("co\n");
}
24
#include<stdio.h>
void nhap(int &n);
void xuatMang(int A[], int n);
void tinhday(int A[], int &n);
void main()
{
    int A[120],n;
    nhap(n);
    tinhday(A,n);
    xuatMang(A,n);
}
void nhap(int &n)
{
    scanf("%d",&n);
}
int tinhfibonacci(int a, int b)

```

```

    nhapMang(A,n);
    int max=timMax(A,n);
    int min=timMin(A,n);
    xuat(max, min);
}
void nhapMang( int A[], int &n)
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%d",&A[i]);
}
int timMax( int A[], int &n)
{
    int max=A[0];
    for(int i=0;i<n;i++)
    {
        if(A[i]>max)
            max=A[i];
    }
    return max;
}
int timMin( int A[], int &n)
{
    int min=A[0];
    for(int i=0;i<n;i++)
    {
        if(A[i]<min)
            min=A[i];
    }
    return min;
}
void xuat(int a, int b)
{
    printf("%d %d",a,b);
}

```

27

```

#include<stdio.h>
void nhapMang( int A[], int &n);
int tim(int A[], int n, int &x);
void xuat(int kq, int x);
void main()
{
    int A[150],n,x;
    nhapMang(A,n);
    int kq=tim(A,n,x);
    xuat(kq,x);
}
void nhapMang( int A[], int &n)
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%d",&A[i]);
}
int tim(int A[], int n, int &x)
{
    for(int i=0;i<n;i++)
    {
        int a,s,x1;

```

```

{
    int s=a+b;
    return s;
}
void tinhday(int A[], int &n)
{
    int a=0;
    int b=1;
    for(int i=2;i<n;i++)
    {
        A[0]=1;
        A[1]=1;
        A[i]=tinhfibonacci(A[i-1],A[i-
2]);
    }
}
void xuatMang(int A[], int n)
{
    for(int i=0;i<n;i++)
        printf("%d ",A[i]);
}

```

26

```

#include<stdio.h>
void nhapMang(int A[], int &n);
int hoandoi( int A[], int n);
void xuatMang( int A[], int n);
void main()
{
    int A[100],n;
    nhapMang(A,n);
    int kq=hoandoi(A,n);
    xuatMang(A,n);
}
void nhapMang(int A[], int &n)
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%d",&A[i]);
}
int timmax( int A[] , int n)
{
    int max=A[0];
    for(int i=0;i<n;i++)
    {
        if(A[i]> max)
        {
            max=A[i];
        }
    }
    return max;
}
int timmin( int A[] , int n)
{
    int min=A[0];
    for(int i=0;i<n;i++)
    {
        if(A[i]< min)

```

```

        a=0;
        s=0;
        x1=A[i];
        while(A[i]!=0 && A[i]>9)
        {
            a=A[i]%10;
            s=(s*10)+a;
            A[i]=A[i]/10;
        }
        if (s==x1)
        {
            x=x1;
            return 1;
        }
    }
    return 0;
}
void xuat(int kq, int x)
{
    if(kq==1)
        printf("%d",x);
    else
        printf("0\n");
}

```

29

```

#include<stdio.h>
void nhapMang( float A[], int &n);
float timA( float A[], int n);
float timB( float A[], int n);
void xuat(float a, float b);
void main()
{
    float A[150];
    int n;
    nhapMang(A,n);
    float a=timA(A,n);
    float b=timB(A,n);
    xuat(a,b);
}
void nhapMang( float A[], int &n)
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%f",&A[i]);
}
float timA( float A[], int n)
{
    float a;
    float min=A[0];
    for(int i=0;i<n;i++)
    {
        if(A[i]<min)
        {
            min=A[i];
        }
    }
    if(min<0)
    {

```

```

        min=A[i];
    }
}
return min;
}
int timvtmax( int A[] , int n)
{
    int max=A[0];
    int vtmax=0;
    for(int i=0;i<n;i++)
    {
        if(A[i]> max)
        {
            max=A[i];
            vtmax=i;
        }
    }
    return vtmax;
}
int timvtmin( int A[] , int n)
{
    int min=A[0];
    int vtmin=0;
    for(int i=0;i<n;i++)
    {
        if(A[i]<min)
        {
            min=A[i];
            vtmin=i;
        }
    }
    return vtmin;
}
int hoandoi( int A[], int n)
{
    int vtmax=timvtmax(A,n);
    int vtmin=timvtmin(A,n);
    int max=timmax(A,n);
    int min=timmin(A,n);
    for(int i=0;i<n;i++)
    {
        if(i==vtmax)
        {
            A[i]=min;
        }
        if(i==vtmin)
        {
            A[i]=max;
        }
    }
    return 1;
}
void xuatMang( int A[], int n)
{
    for(int i=0;i<n;i++)
        printf("%d ",A[i]);

```

```

        a=int(min)-1;
    }
    else
        a=int(min)-1;
    return a;
}
float timB( float A[], int n)
{
    float b;
    float max=A[0];
    for(int i=0;i<n;i++)
    {
        if(A[i]>max)
        {
            max=A[i];
        }
    }
    if(max<0)
    {
        b=int(max)+1;
    }
    else
        b=int(max)+1;
    return b;
}
void xuat(float a, float b)
{
    printf("%.0f %.0f",a,b);
}
30
#include<stdio.h>
void nhap( float A[], int &nA, float &x);
void kt( float A[], int nA, float x);
void main()
{
    float A[150],x;
    int nA;
    nhap(A,nA,x);
    kt(A,nA,x);
}
void nhap( float A[], int &nA, float &x)
{
    scanf("%d",&nA);
    for( int i=0;i<nA;i++)
        scanf("%f",&A[i]);
    scanf("%f",&x);
}
void kt( float A[], int nA, float x)
{
    int flag=0;
    float a;
    for( int i=nA-1;i>=0;i--)
    {
        if( A[i] >x && A[i] <0 )
        {
            flag=1;
            a=A[i];
            break;

```

```

}
28
#include<stdio.h>
void nhapMang(int A[], int &n);
int timkiem(int A[], int n, int &x);
void xuat(int kq, int x);
void main()
{
    int A[150],n,x;
    nhapMang(A,n);
    int kq=timkiem(A,n,x);
    xuat(kq,x);
}
void nhapMang(int A[], int &n)
{
    scanf("%d",&n);
    for(int i=0;i<n;i++)
        scanf("%d",&A[i]);
}
int timkiem(int A[], int n, int &x)
{
    for(int i=0;i<n;i++)
    {
        int a;
        a=A[i]%10;
        if((a%2)!=0)
        {
            x=A[i];
            return 1;
        }
    }
    return 0;
}
void xuat(int kq, int x)
{
    if(kq==1)
        printf("%d ",x);
    else
        printf("0 ");
}

```

<pre> } } if(flag==1) printf("%.2f",a); else printf("0"); }</pre>	
--	--