

TRƯỜNG ĐẠI HỌC ĐIỆN LỰC
KHOA CÔNG NGHỆ THÔNG TIN



ĐẠI HỌC ĐIỆN LỰC
ELECTRIC POWER UNIVERSITY

TÀI LIỆU THỰC HÀNH
MÔN: NHẬP MÔN TIN HỌC
(Phiên bản 1, lưu hành nội bộ)

Hà Nội, 2018

MỘT SỐ BÀI TẬP CƠ BẢN MÔN NHẬP MÔN TIN HỌC

PHẦN 1: NHẬP, XUẤT VÀ CÁC CÂU LỆNH CÓ CẤU TRÚC

Bài 1.

Nhập vào 2 số a, b. Tính tổng, hiệu, tích, thương của 2 số a và b.

* Thuật toán:

- + Tính $a+b$, $a-b$, $a*b$
- + Nếu b bằng 0, hiển thị “mẫu số bằng 0”
- Ngược lại, tính a/b

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    float a, b, tong, hieu, tich, thuong;
    printf("Nhap 2 so: ");
    scanf("%f%f", &a, &b);
    tong = a+b;
    hieu = a-b;
    tich = a*b;
    printf("a+b=%f, a-b=%f, a*b=%f, ", tong, hieu, tich);
    if(b==0)
    {
        printf("Mau so bang 0");
    }
    else
    {
        thuong = a/b;
        printf("a/b=%f", thuong);
    }
}
```

Bài 2.

Giải phương trình bậc nhất 1 ẩn ($ax+b=0$).

* Thuật toán:

- + Xét trường hợp $a = 0$, sau đó xét $b = 0$ để xét các trường hợp của PT.

* Chương trình:

```
#include <conio.h>
```

```
#include <stdio.h>
int main()
{
    float a, b;
    printf("Nhap 2 he so cua PT bac nhat: ");
    scanf("%f%f", &a, &b);
    if(a==0)
    {
        if(b==0)
            printf("PT vo so nghiem!");
        else
            printf("PT vo nghiem!");
    }
    else
    {
        printf("PT co nghiem x = %f", -b/a);
    }
}
```

Bài 3.

Kiểm tra một số có phải số chính phương không (sử dụng câu lệnh rẽ nhánh).

* Thuật toán:

+ Lấy phần nguyên của phép \sqrt{a} .

+ so sánh a với bình phương của phần nguyên đó.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
int main()
{
    int a;
    printf("Nhap 1 so nguyen: ");
    scanf("%d", &a);
    int b = (int)sqrt(a);
    if(b*b == a)
    {
        printf("%d la so chinh phuong", a);
    }
    else
    {
        printf("%d khong phai so chinh phuong", a);
    }
}
```

Bài 4.

Tính TBC các số chẵn từ 2 đến n (n nhập từ bàn phím).

* Thuật toán:

- + Bắt đầu từ vị trí số 2, mỗi lần lặp tăng 2 đơn vị
- + Qua mỗi vị trí:
 - . Cộng dồn giá trị vào Tổng.
 - . Tăng biến đếm lên 1 đơn vị.
- + Nếu $Dem > 0$, $Tính\ TBC = Tong / Dem$
- Ngược lại, thông báo không có số chẵn.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int n, Tong;
    printf("Nhap so nguyen n=: ");
    scanf("%d", &n);
    Tong = 0;
    Dem = 0;
    for(int i=2; i<=n; i=i+2)
    {
        Tong += i;
        Dem++;
    }
    if(Dem>0)
    {
        printf("Tong = %d", Tong);
    }
    else
    {
        printf("Khong co so chan.");
    }
}
```

Bài 5.

Nhập vào từ bàn phím số nguyên không âm n, tính n!.

* Thuật toán:

- + nhập liên tiếp n đến khi n thỏa mãn $n \geq 0$

- + khởi tạo giaithua = 1
- + Duyệt từ 1 đến n
- + Tại mỗi vị trí, nhân dồn giá trị đang xét với giaithua.

*** Chương trình:**

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int n, giaithua;
    do
    {   printf("nhap so can tinh giai thua: ");
        scanf("%d", &n);
    }
    while (n<=0);
    giaithua = 1;
    for(int i=1;i<=n;i++)
    {
        giaithua *= i;
    }
    printf("%d! = %d",n,giaithua);
}
```

Bài 6.

Nhập vào số dòng n, hiển thị lên màn hình hình sau:

Ví dụ n = 4

```
*
**
***
****
```

*** Thuật toán:**

- + Xét từ hàng 1 đến n
- + Tại mỗi hàng thứ i, hiển thị i ký tự '*' và 1 dấu xuống dòng.

*** Chương trình:**

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int n;
```

```
printf("Nhap so nguyen n=: ");
scanf("%d", &n);
for(int i=1;i<=n;i++)
{
    for(int j=1;j<=i;j++)
    {
        printf("*");
    }
    printf("\n");
}
```

Bài 7.

Nhập vào số dòng n, hiển thị lên màn hình hình sau:

Ví dụ n = 4

```
*****
****
***
**
*
```

* Thuật toán:

+ Xét từ hàng 1 đến n

+ Tại mỗi hàng thứ i, hiển thị (n-i+1) ký tự '*' và 1 dấu xuống dòng.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int n;
    printf("Nhap so nguyen n=: ");
    scanf("%d", &n);
    for(int i=1;i<=n;i++)
    {
        for(int j=1;j<=n-i+1;j++)
        {
            printf("*");
        }
        printf("\n");
    }
}
```

Bài 8.

Nhập vào số dòng m, số cột n, hiển thị lên màn hình hình sau(m dòng mỗi dòng n dấu *):

Ví dụ m = 3, n = 4

* Thuật toán:

+ Xét từ hàng 1 đến m

+ Tại mỗi hàng thứ i, hiển thị n ký tự '*' và 1 dấu xuống dòng.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int n, m;
    printf("Nhap 2 so nguyen: ");
    scanf("%d%d", &m, &n);
    for(int i=1; i<=m; i++)
    {
        for(int j=1; j<=n; j++)
        {
            printf("*");
        }
        printf("\n");
    }
}
```

Bài 9.

Nhập vào từ bàn phím 2 số a và b (a,b>0), rút gọn phân số a/b

* Thuật toán:

+ Tìm UCLN của 2 số a và b.

+ Chia a và b cho UCLN, hiển thị thương rút gọn lên MH.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int a, b;
```

```
do
{   printf("Nhap 2 so nguyen: ");
    scanf("%d%d", &a, &b);
}
while(a<=0||b<=0);
int UCLN;
for(int i=a;i>=1;i--)
{
    if(a%i == 0 && b%i == 0)
    {
        UCLN = i;
        break;
    }
}
a = a/UCLN;
b = b/UCLN;
printf("Phan so sau rut gon : %d/%d", a, b);
}
```

Bài 10.

Đếm số ước số của số nguyên N.

* Thuật toán:

+ Duyệt từ 1 đến N

+ Tại vị trí đang xét, nếu N chia hết cho giá trị tại vị trí đang xét, tăng biến

Dem lên 1.

Ngược lại, chuyển qua vị trí kế tiếp

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int n;
    do
    {   printf("Nhap so nguyen n=: ");
        scanf("%d", &n);
    }
    while(n<=0);
    int Dem = 0;
    for(int i=1;i<=n;i++)
    {
        if(n%i==0)
        {
```



```
        Dem += 1;
    }
}
printf("Số ước = %d", Dem);
}
```

Bài 11.

Giải phương trình bậc hai 1 ẩn số $ax^2 + bx + c = 0$

* Thuật toán:

+ Lần lượt xét $a = 0$, $b = 0$ rồi xét $c = 0$ để xét các trường hợp của phương trình.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
int main()
{
    float a, b, c, x1, x2;
    printf("nhap cac he so a,b,c: ");
    scanf("%f%f%f", &a, &b, &c);
    if(a == 0)
    {
        if(b==0)
        {
            if(c==0)
            {
                printf("PT co vo so nghiem!");
            }
            else
            {
                printf("PT vo nghiem!");
            }
        }
        else
        {
            x1 = -c/b;
            printf("PT co nghiem duy nhat x = %f", x1);
        }
    }
    else
    {
        float delta = b*b-4*a*c;
        if(delta < 0)
        {
```

```
        printf("PT vo nghiem!");  
    }  
    else  
    {  
        if(delta == 0)  
        {  
            x1 = -b/(2*a);  
            printf("PT co nghiem x = %f", x1);  
        }  
        else  
        {  
            x1 = (-b - sqrt(delta))/(2*a);  
            x2 = (-b + sqrt(delta))/(2*a);  
            printf("PT co 2 nghiem phan biet x1 =  
%f\nx2 = %f", x1, x2);  
        }  
    }  
}
```

Bài 12.

Nhập vào 3 số a, b và c. Kiểm tra a, b và c có phải độ dài 3 cạnh của tam giác hay không. Nếu đúng tính chu vi và diện tích tam giác.

* Thuật toán:

+ Sử dụng bất đẳng thức tam giác, kiểm tra điều kiện thỏa mãn của tam giác.

+ Nếu thỏa mãn, thông báo là tam giác, tính chu vi, diện tích (công thức Heron) $S = \sqrt{p(p-a)(p-b)(p-c)}$ với p là nửa chu vi.

Ngược lại, thông báo không phải là 3 cạnh của tam giác

* Chương trình:

```
#include <conio.h>  
#include <stdio.h>  
#include <math.h>  
int main()  
{  
    float a, b, c, CV, S;  
    printf("nhap 3 so a,b,c: ");  
    scanf("%f%f%f", &a, &b, &c);  
    if(a>0 && b>0 && c>0 && a+b>c && a+c>b && c+b>a)  
    {  
        printf("Thoa man la tam giac!\n");  
        CV = a+b+c;  
        float p = CV/2;
```

```
        S = sqrt(p*(p-a)*(p-b)*(p-c));  
        printf("Chu vi : %f\nDien tich : %f", CV, S);  
    }  
    else  
    {  
        printf("Khong thoa man la tam giac!");  
    }  
}
```

Bài 13.

Nhập vào 2 số a và b. Kiểm tra a, b có phải độ dài 2 cạnh của hình vuông hay không? Nếu có tính chu vi và diện tích của hình vuông.

* Thuật toán:

- + Kiểm tra a, b dương và tương quan giữa a và b.
- + Nếu a bằng b, thông báo tứ giác là hình vuông, tính chu vi và diện tích, hiển thị lên màn hình.

Ngược lại, thông báo không phải là độ dài 2 cạnh của hình vuông.

* Chương trình:

```
#include <conio.h>  
#include <stdio.h>  
int main()  
{  
    float a, b, CV, S;  
    printf("nhap 2 so a,b: ");  
    scanf("%f%f", &a, &b);  
    if(a>0 && b>0 && a==b)  
    {  
        printf("Thoa man do dai 2 canh hinh vuong!\n");  
        CV = 4*a;  
        S = a*a;  
        printf("Chu vi : %f\nDien tich : %f", CV, S);  
    }  
    else  
    {  
        printf("Khong thoa man la hinh vuong!");  
    }  
}
```

Bài 14.

Nhập vào 1 số tự nhiên có 3 chữ số ($99 < n < 1000$), tính tổng lập phương của các chữ số của số vừa nhập, hiển thị lên màn hình.

* Thuật toán:

- + Nhập vào giá trị của n, kiểm tra n thỏa mãn $99 < n < 1000$
- + Chia lấy phần dư của n cho 10 để tách từng chữ số của n, cộng dồn tổng lập phương của chữ số với Tong.
- + Gán $n = n/10$ để xét đến chữ số kế tiếp
- + Thực hiện đến khi $n = 0$.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int n;
    do
    {
        printf("Nhap n (99<n<1000) = ");
        scanf("%d", &n);
    }
    while(n<=99 || n>= 1000);
    int Tong = 0;
    while(n!=0)
    {
        int a = n%10;
        n = n/10;
        Tong += a*a*a;
    }
    printf("Tong = %d", Tong);
}
```

Bài 15.

Nhập vào 1 số nguyên dương n ($n > 100$), đảo ngược các chữ số của số vừa nhập.

Ví dụ: 123 => 321

* Thuật toán:

- + Nhập vào giá trị của n, kiểm tra n thỏa mãn $n > 100$
- + Chia lấy phần dư của n cho 10 để tách từng chữ số của n, nhân số _đảo lên 10 rồi cộng thêm giá trị của chữ số vừa tách được.
- + Gán $n = n/10$ để xét đến chữ số kế tiếp
- + Thực hiện đến khi $n = 0$.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int n;
    do
    {   printf("Nhap n = ");
        scanf("%d", &n);
    }
    while (n<=0);
    int n_dao = 0;
    while (n!=0)
    {
        int du = n%10;
        n_dao = n_dao *10 + du;
        n = n/10;
    }
    printf("So dao : %d", n_dao);
}
```

Bài 16.

Kiểm tra một số có phải là số hoàn hảo không

+ Số hoàn hảo là số có tổng các ước nhỏ hơn nó bằng chính nó.

Ví dụ: 6 có 3 ước (các ước <6) là 1, 2 và 3; có $1 + 2 + 3 = 6$ nên 6 là số hoàn hảo.

* Thuật toán:

+ Duyệt các số từ 1 đến N-1, nếu là ước thì cộng thêm vào tổng.

+ Nếu tổng các ước bằng N thì thông báo là số hoàn hảo

Ngược lại, thông báo không phải số hoàn hảo.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int n, Tong;
    printf("Nhap n = ");
    scanf("%d", &n);
    Tong = 0;
    for(int i=1; i<n; i++)
    {
        if (n%i==0)
```

```
        {
            Tong += i;
        }
    }
    if (Tong == n)
    {
        printf("%d la so hoan hao!", n);
    }
    else
    {
        printf("%d khong la so hoan hao!", n);
    }
}
```

Bài 17.

Nhập vào từ bàn phím số nguyên dương n , tính $n!!$

$$n!! = \begin{cases} 1 * 3 * 5 * \dots * n & (\text{nếu } n \text{ lẻ}) \\ 2 * 4 * 6 * \dots * n & (\text{nếu } n \text{ chẵn}) \end{cases}$$

* Thuật toán:

+ Kiểm tra nhập số nguyên dương n .

+ Nếu n lẻ, nhân liên tiếp các số lẻ

Ngược lại, nhân liên tiếp các số chẵn

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int n, GTT;
    do
    {
        scanf("%d", &n);
    }
    while (n <= 0);
    GTT = 1;
    int i;
    if (n % 2 == 0)
    {
        for (int i = 2; i <= n; i = i + 2)
        {
            GTT *= i;
        }
    }
}
```

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int n, Tong;
    printf("Nhap n = ");
    scanf("%d", &n);
    Tong = 0;
    for (int i = 1; i < n; i++)
    {
        if (n % i == 0)
```

```
else
{
    for(int i=1;i<=n;i=i+2)
    {
        GTT *= i;
    }
}
printf("%d!! = %d", n, GTT);
}
```

Bài 18.

Kiểm tra một số có phải là số nguyên tố không?

* Thuật toán:

- + Đếm các ước trong khoảng (1, n)
- + Nếu số lượng ước số bằng 0, thông báo là số nguyên tố
- Ngược lại, thông báo không phải số nguyên tố

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int n, GTT;
    do
    {
        printf("Nhap n = ");
        scanf("%d", &n);
    }
    while(n<=0);
    int SL_Uoc = 0;
    for(int i=2;i<n;i++)
    {
        if (n%i==0)
        {
            SL_Uoc++;
        }
    }
    if(SL_Uoc == 0)
    {
        printf("%d la so nguyen to!", n);
    }
    else
    {
        printf("%d khong phai so nguyen to!", n);
    }
}
```

}

Bài 19.

Hiển thị tất cả các số nguyên tố trong đoạn $[1, n]$

* Thuật toán:

+ Duyệt qua từng vị trí từ 1 đến n .

+ Kiểm tra số đang duyệt có phải số nguyên tố không, nếu đúng hiển thị giá trị ra màn hình

Ngược lại, chuyển đến vị trí tiếp theo.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int n, GTT;
    do
    {
        printf("Nhap n = ");
        scanf("%d", &n);
    }
    while(n<=0);
    for(int i=1;i<=n;i++)
    {
        int SL_Uoc = 0;
        for(int j=2;j<i;j++)
        {
            if (i%j==0)
            {
                SL_Uoc++;
            }
        }
        if(SL_Uoc == 0)
        {
            printf("%5d", i);
        }
    }
}
```

Bài 20.

Nhập vào 2 số nguyên dương a và b , hiển thị UCLN và BCNN của 2 số a, b

* Thuật toán:

+ Kiểm tra việc nhập 2 số nguyên dương.

+ Tìm UCLN của 2 số

C1: Duyệt từ số nhỏ hơn trở về 1, nếu gặp ước của cả 2 số thì ghi nhận và dừng lại.

C2: Dùng thuật giải tìm UCLN

+ BCNN = (a*b)/UCLN.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int a, b, UCLN, BCNN;
    do
    {
        printf("Nhap n = ");
        scanf("%d%d", &a, &b);
    }
    while(a<=0 || b<=0);
    for(int i=a;i>=1;i--)
    {
        if(a%i==0 && b%i==0)
        {
            UCLN = i;
            break;
        }
    }
    BCNN = (a*b)/UCLN;
    printf("UCLN : %d\nBCNN : %d", UCLN, BCNN);
}
```

Bài 21.

Nhập vào từ bàn phím số nguyên dương n, tính và hiển thị giá trị biểu thức

$$f(n) = 1 - 2 + 3 - 4 + \dots + (-1)^{n+1} * n$$

* Thuật toán:

+ Kiểm tra nhập số nguyên dương n.

+ Duyệt liên tiếp từ 1 đến n, cộng dồn giá trị của biểu thức.

* Chương trình:

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

```
int main()
{
    int n, f;
    do
    {   printf("Nhap n = ");
        scanf("%d", &n);
    }
    while(n<=0);
    f = 0;
    int k = -1;
    for(int i=1; i<=n; i++)
    {
        k = k * (-1);
        f = f + k * i;
    }
    printf("f(%d) = %d", n, f);
}
```

Bài 22.

Nhập vào từ bàn phím số nguyên n, tính và hiển thị giá trị của biểu thức sau:

$$f(n) = \begin{cases} 0 & \text{nếu } n \leq 0 \\ \frac{2^1}{1!} + \frac{2^2}{2!} + \frac{2^3}{3!} + \dots + \frac{2^n}{n!} & \text{nếu } n > 0 \end{cases}$$

* Thuật toán:

- + Nếu $n \leq 0$, hiển thị $f(n) = 0$
- + Ngược lại, duyệt từ 1 đến n
- + Tại mỗi vị trí, tính mũ (thông qua hàm exp), và giai thừa (nhân liên tiếp các giá trị).

* Chương trình:

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

int main()
{
    int n;
    printf("Nhap n = ");
    scanf("%d", &n);
    float Fn = 0;
    if(n<=0)
    {
        printf("F(%d) = 0", n);
    }
}
```

```
    }  
    else  
    {  
        int ts = 1;  
        int ms = 1;  
        for(int i=1;i<=n;i++)  
        {  
            ts = ts * 2;  
            ms = ms * i;  
            Fn = Fn + (float)ts/ms;  
        }  
        printf("F(%d) = %f", n, Fn);  
    }  
}
```

Bài 23.

Nhập vào số nguyên dương trong đoạn [0, 9], hiển thị lên màn hình tiếng anh của số đó.

* Thuật toán:

+ Kiểm tra giá trị nhập từ bàn phím

* Chương trình:

```
#include <conio.h>  
#include <stdio.h>  
#include <math.h>  
int main()  
{  
    int n;  
    do  
    {    printf("Nhap n = ");  
        scanf("%d", &n);  
    }  
    while(n<0 || n>9);  
    switch(n)  
    {  
        case 0: printf("Zero"); break;  
        case 1: printf("One"); break;  
        case 2: printf("Two"); break;  
        case 3: printf("Three"); break;  
        case 4: printf("Four"); break;  
        case 5: printf("Five"); break;  
        case 6: printf("Six"); break;  
        case 7: printf("Seven"); break;  
        case 8: printf("Eight"); break;  
        case 9: printf("Nine"); break;  
    }  
}
```

```
}  
}
```

Bài 24.

Tính tiền điện phải trả cho 1 hộ dân biết:

+ Số tiền trong định mức: 1000 VNĐ/kWh

+ Số tiền ngoài định mức: 2000 VNĐ/kWh

Trong đó định mức đối với mỗi mục đích sử dụng khác nhau là:

1. Hộ gia đình: 100 kWh

2. Hộ kinh doanh: 250kWh

3. Hộ sản xuất: 420 kWh

Nhập vào từ bàn phím số mục đích sử dụng(dưới dạng số thứ tự) của hộ và số điện, hiển thị lên màn hình số tiền hộ đó phải trả trong tháng.

* Thuật toán:

+ Sử dụng hằng lưu các giá trị định mức

+ Đọc vào số điện và số thứ tự của hộ.

+ Lấy giá trị định mức dựa theo số thứ tự.

+ Nếu số điện vượt định mức, tính tiền theo công thức $DM \cdot 1000 + dư \cdot 2000$

Ngược lại số tiền = số điện * 1000.

* Chương trình:

```
#include <conio.h>  
#include <stdio.h>  
#define TDM 1000  
#define NDM 2000  
#define HGD 100  
#define HKD 250  
#define HSX 420  
int main()  
{  
    int n, STT;  
    int SoTien;  
    printf("Nhap so dien ho da su dung: n = ");  
    do  
    {  
        scanf("%d", &n);  
    }  
    while(n<0);  
    printf("Nhap loai ho:\n1. Ho gia dinh\n2. Ho kinh  
doanh\n3. Ho san xuat\n");
```

```
do
{
    scanf("%d", &STT);
}
while(STT<1 || STT>3);
int DM;
switch(STT)
{
    case 1: DM = HGD; break;
    case 2: DM = HKD; break;
    case 3: DM = HSX; break;
}
if(n - DM > 0)
{
    SoTien = DM * TDM + (n-DM) * NDM;
}
else
{
    SoTien = n * TDM;
}
printf("So tien phai tra : %d", SoTien);
}
```

Bài 25.

Nhập vào từ bàn phím số nguyên dương n, hiển thị lên màn hình dạng sau:

VD: n = 4

```
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 16.
```

* Thuật toán:

- + Kiểm tra số nhập vào là nguyên dương.
- + Duyệt từ 1 đến n^2 , hiển thị giá trị đang xét
- + Nếu vị trí đang xét chia hết cho n, hiển thị một dấu xuống dòng ("\n"),
Ngược lại, chuyển đến vị trí kế tiếp.

* Chương trình:

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

```
int n;  
do  
{   printf("Nhap n = ");  
    scanf("%d", &n);  
}  
while (n<=0);  
for(int i=1;i<=n*n;i++)  
{  
    printf("%5d", i);  
    if(i%n == 0)  
    {  
        printf("\n");  
    }  
}  
}
```

Bài 26.

Dãy số Fibonacci được tạo ra như sau:

$F(1) = 1; F(2) = 1$

$F(n) = F(n-1) + F(n-2)$

Nhập vào từ bàn phím số nguyên n, hiển thị dãy fibonacci từ 1 đến n.

* Thuật toán:

- + khởi tạo $a = 1, b = 1$;
- + Duyệt liên tiếp từ 1 đến n,
- + Nếu $i = 1$ hoặc $i = 2$, hiển thị giá trị 1.

Ngược lại, tính $c = a + b$ ($f(n) = f(n-1) + f(n-2)$), cập nhật lại giá trị của a và b.

* Chương trình:

```
#include <conio.h>  
#include <stdio.h>  
int main()  
{  
    int n;  
    do  
    {   printf("Nhap n = ");  
        scanf("%d", &n);  
    }  
    while (n<=0);  
    int a = 1, b = 1;  
    for(int i=1;i<=n;i++)  
    {  
        if(i == 1 || i == 2)
```

```
        {
            printf("%5d", 1);
        }
    else
    {
        int c = a + b;
        printf("%5d", c);
        a = b;
        b = c;
    }
}
```

Bài 27.

Nhập vào từ bàn phím các giá trị của 2 đường tròn (x_1, y_1, R_1) và (x_2, y_2, R_2) .

Kiểm tra xem 2 đường tròn có tiếp xúc không?

* Thuật toán:

+ Tính khoảng cách giữa 2 tâm $O_1(x_1, y_1)$ và $O_2(x_2, y_2)$.

+ Nếu khoảng cách tâm = $R_1 + R_2$ hoặc bằng $\text{abs}(R_1 - R_2)$ thông báo 2 đường tròn tiếp xúc.

Ngược lại, thông báo 2 đường tròn không tiếp xúc.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
int main()
{
    float x1, y1, R1, x2, y2, R2;
    printf("Nhap thong tin duong tron thu 1 : ");
    scanf("%f%f%f", &x1, &y1, &R1);
    printf("Nhap thong tin duong tron thu 2 : ");
    scanf("%f%f%f", &x2, &y2, &R2);
    float KC = sqrt(pow(x1-x2,2)+pow(y1-y2,2));
    if(KC == R1+R2 || KC == fabs(R1-R2))
    {
        printf("2 duong tron tiep xuc!");
    }
    else
    {
        printf("2 duong tron khong tiep xuc!");
    }
}
```

PHẦN 2: MẢNG

Bài 28.

Nhập vào từ bàn phím mảng số nguyên gồm n phần tử. Hiển thị mảng và mảng đảo lên màn hình?

* Thuật toán:

+ Với trường hợp mảng đảo của mảng đã nhập, duyệt từ cuối mảng về đầu, mỗi lần giảm 1 đơn vị.

+ Tại mỗi vị trí, hiển thị giá trị lên màn hình.

* Chương trình:

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

int main()
{
    int a[100], n;
    printf("Nhap so phan tu n = ");
    scanf("%d", &n);
    for(int i=0; i<n; i++)
    {
        printf("a[%d] = ", i);
        scanf("%d", &a[i]);
    }
    printf("\nMang sau khi nhap la : ");
    for(int i=0; i<n; i++)
    {
        printf("%5d", a[i]);
    }
    printf("\nMang dao cua mang da nhap la : ");
    for(int i=n-1; i>=0; i--)
    {
        printf("%5d", a[i]);
    }
}
```

Bài 29.

Nhập từ bàn phím mảng số thực gồm n phần tử, hiển thị mảng lên màn hình, tính giá trị trung bình của mảng.

* Thuật toán:

+ Duyệt từ đầu đến cuối mảng

+ Tại mỗi vị trí, cộng dồn giá trị của mảng với Tong.

+ Tính TBC = Tong / n

*** Chương trình:**

```
#include <conio.h>
#include <stdio.h>
int main()
{
    float a[50];
    int n;
    scanf("%d", &n);
    for(int i=0;i<n;i++)
    {
        scanf("%f", &a[i]);
    }
    printf("Mang vua nhap la : ");
    for(int i=0;i<n;i++)
    {
        printf("%5f", a[i]);
    }
    if(n == 0)
    {
        printf("Mang khong co phan tu");
    }
    else
    {
        float Tong = 0;
        for(int i=0;i<n;i=i+1)
        {
            Tong += a[i];
        }
        printf("TBC = %f",Tong/n);
    }
}
```

Bài 30.

Nhập từ bàn phím mảng số nguyên gồm n phần tử, tính trung bình cộng các phần tử lẻ ở vị trí chẵn.

*** Thuật toán:**

+ Bắt đầu từ vị trí 0, mỗi lần duyệt tăng 2 đơn vị.

+ Tại mỗi vị trí, nếu giá trị lẻ (không chia hết cho 2), cộng dồn giá trị tại vị trí này cho Tong, tăng biến Dem lên 1 đơn vị.

Ngược lại, chuyển đến vị trí kế tiếp.

+ Nếu $Dem > 0$, tính $TBC = Tong / Dem$

Ngược lại, thông báo không có số lẻ ở vị trí chẵn

*** Chương trình:**

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int a[50], n;
    scanf("%d", &n);
    for(int i=0;i<n;i++)
    {
        scanf("%d", &a[i]);
    }
    printf("Mang vua nhap la : ");
    for(int i=0;i<n;i++)
    {
        printf("%5d", a[i]);
    }
    int Tong = 0, Dem = 0;
    for(int i=0;i<n;i=i+2)
    {
        if(a[i]%2 == 1)
        {
            Tong += a[i];
            Dem++;
        }
    }
    if(Dem > 0)
    {
        float TBC = (float)Tong / Dem;
        printf("TBC = %f", TBC);
    }
    else
    {
        printf("Khong co phan tu le o vi tri chan!");
    }
}
```

Bài 31.

Nhập từ bàn phím mảng số nguyên gồm n phần tử, hiển thị vị trí của các phần tử chia hết cho 3 có trong mảng.

*** Thuật toán:**

+ Duyệt liên tiếp từ 0 đến $n - 1$

+ Tại mỗi vị trí, nếu giá trị của mảng tại vị trí này chia hết cho 3 thì hiển thị vị trí đang xét lên màn hình.

Ngược lại, chuyển đến vị trí kế tiếp.

*** Chương trình:**

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int a[50], n;
    scanf("%d", &n);
    for(int i=0;i<n;i++)
    {
        scanf("%d", &a[i]);
    }
    printf("Mang vua nhap la : ");
    for(int i=0;i<n;i++)
    {
        printf("%5d", a[i]);
    }
    printf("Vi tri cua cac phan tu chia het cho 3: ");
    for(int i=0;i<n;i=i+1)
    {
        if(a[i]%3 == 0)
        {
            printf("%5d", i);
        }
    }
}
```

Bài 32.

Nhập vào từ bàn phím mảng nguyên gồm n phần tử, xóa phần tử âm có trong mảng.

*** Thuật toán:**

Duyệt từ vị trí đầu tiên đến vị trí cuối của mảng:

Nếu có phần tử <0 thì đó là phần tử cần xóa

- +Thực hiện theo phương pháp dồn toa để lấp vị trí cần xóa
- + Duyệt từ vị trí cần xóa, chuyển đến sát giá trị cuối của mảng.
- + Sau khi kết thúc, giảm số lượng phần tử đi 1.

*** Chương trình:**

```
#include <conio.h>
```

```
#include <stdio.h>
int main()
{
    int a[50], n;
    scanf("%d", &n);
    for(int i=0;i<n;i++)
    {
        scanf("%d", &a[i]);
    }
    printf("Mang vua nhap la : ");
    for(int i=0;i<n;i++)
    {
        printf("%5d", a[i]);
    }
    for(i=0;i<n;i++)
        if (a[i]<0)
            {for(int j=i;j<n-1;j++)
                a[j]=a[j+1];
                n=n-1;
            }
    printf("\n Day so sau khi xoa la:");
    for(i=0;i<n;i++)
        printf("%5d", a[i]);
}
```

Bài 33.

Nhập vào từ bàn phím mảng nguyên gồm n phần tử, thêm 1 phần tử x vào vị trí thứ k của mảng.

* Thuật toán:

+ Thực hiện theo phương pháp dồn toa để tạo một toa trống cho phép chèn thêm vị trí vào:

+ Duyệt từ cuối về vị trí cần chèn, chuyển giá trị đến vị trí kế tiếp.

+ Sau khi kết thúc, chèn giá trị x vào vị trí trống, tăng số lượng phần tử lên 1.

* Chương trình:

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

int main()
{
    int a[100], n;
    int x, k;
    printf("So luong phan tu : ");
```

```
scanf("%d", &n);
for(int i=0;i<n;i++)
{
    printf("a[%d] = ", i);
    scanf("%d", &a[i]);
}
printf("Nhap gia tri can chen x = ");
scanf("%d", &x);
printf("Nhap vi tri chen k = ");
scanf("%d", &k);
for(int i=n;i>k;i--)
{
    a[i] = a[i-1];
}
a[k] = x;
n++;
printf("Mang sau khi them: ");
for(int i=0;i<n;i++)
{
    printf("%5d", a[i]);
}
}
```

Bài 34.

Nhập vào từ bàn phím mảng nguyên gồm n phần tử, sắp xếp mảng vừa nhập theo chiều tăng dần.

* Thuật toán:

+ Thực hiện theo ý tưởng thuật toán sắp xếp nổi bọt (đưa giá trị nhỏ nhất lên đầu, và lặp lại đến hết dãy).

+ Duyệt i từ 1 đến n-1

+ Duyệt j từ i+1 đến n

+ Nếu $a[i] > a[j]$ (vị trí trước lớn hơn vị trí sau) thì đổi chỗ giá trị.

Ngược lại, chuyển đến vị trí kế tiếp.

* Chương trình:

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

int main()
{
    int a[100], n, tg;
    printf("Nhap so luong phan tu : ");
```

```
scanf("%d", &n);
for(int i=0;i<n;i++)
{
    printf("a[%d] = ", i);
    scanf("%d", &a[i]);
}
for(int i=0;i<n-1;i++)
{
    for(int j=i+1;j<n;j++)
    {
        if(a[i]>a[j])
        {
            tg = a[i];
            a[i] = a[j];
            a[j] = tg
        }
    }
}
printf("Mang sap xep: ");
for(int i=0;i<n;i++)
{
    printf("%5d", a[i]);
}
}
```

Bài 35.

Nhập vào từ bàn phím mảng nguyên gồm n phần tử, tìm giá trị lớn nhất có trong mảng.

* Thuật toán:

- + Thực hiện theo ý tưởng thuật toán chia để trị.
- + Max được gán bằng a[0].
- + Duyệt lần lượt các vị trí từ 1 đến n, nếu gặp giá trị nào lớn hơn Max thì cập nhật lại giá trị của Max.

* Chương trình:

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

int main()
{
    int a[100], n, max;
    printf("Nhap so luong phan tu : ");
    scanf("%d", &n);
```

```
for(int i=0;i<n;i++)
{
    printf("a[%d] = ", i);
    scanf("%d", &a[i]);
}
max = a[0];
for(int i=1;i<n;i++)
{
    if(a[i]>max)
        max = a[i];
}
printf("Max = %d", max);
}
```

Bài 36.

Nhập vào từ bàn phím mảng số nguyên gồm n phần tử. Tìm vị trí của các phần tử nhỏ nhất trong mảng.

* Thuật toán:

- + Tìm giá trị Min có trong mảng.
 - + Duyệt qua từng vị trí của mảng.
 - + Tại mỗi vị trí, nếu giá trị bằng Min thì hiển thị vị trí đang duyệt;
- Ngược lại, chuyển đến vị trí tiếp theo..

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int a[50], n;
    printf("Nhap so luong phan tu : ");
    scanf("%d", &n);
    for(int i=0;i<n;i++)
    {
        printf("a[%d] = ", i);
        scanf("%d", &a[i]);
    }

    int Min = a[0];
    for(int i=1;i<n;i++)
        if(Min > a[i])

        Min = a[i];
}
```

```
printf("\nVi tri cua cac phan tu Min trong mang : ");  
for(int i=0;i<n;i++)  
{  
    if(a[i] == Min)  
    {  
        printf("%5d", i);  
    }  
}
```

Bài 37.

Nhập vào từ bàn phím mảng số thực gồm n phần tử. Tìm tích các số dương có trong mảng.

* Thuật toán:

- + Khởi tạo Tích = 1.
 - + Duyệt qua từng vị trí của mảng.
 - + Tại mỗi vị trí, nếu giá trị tại vị trí này > 0 thì nhân dồn vào Tích
- Ngược lại, chuyển đến vị trí kế tiếp.

* Chương trình:

```
#include <conio.h>  
#include <stdio.h>  
int main()  
{  
    float a[50];  
    int n;  
    printf("Nhap so luong phan tu : ");  
    scanf("%d", &n);  
    for(int i=0;i<n;i++)  
    {  
        printf("a[%d] = ", i);  
        scanf("%f", &a[i]);  
    }  
    printf("Mang vua nhap la : ");  
    for(int i=0;i<n;i++)  
    {  
        printf("%f", a[i]);  
    }  
    float Tich = 1;  
    for(int i=0;i<n;i++)  
    {  
        if(a[i]>0)
```



```
        {
            Tich *= a[i];
        }
    }
    printf("Tich cac so duong : %f", Tich);
}
```

Bài 38.

Nhập vào từ bàn phím mảng số nguyên gồm n phần tử. Đếm và hiển thị các số chính phương có trong mảng.

* Thuật toán:

- + Khởi tạo Dem = 0
- + Duyệt từ đầu đến cuối mảng
- + Tại mỗi vị trí, nếu giá trị tại vị trí đang xét là số chính phương, tăng biến

Dem lên 1, hiển thị giá trị lên màn hình

Ngược lại, chuyển đến vị trí tiếp theo.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
#include <math.h>
int main()
{
    int a[50], n;
    printf("Nhap so luong phan tu : ");
    scanf("%d", &n);
    for(int i=0;i<n;i++)
    {
        printf("a[%d] = ", i);
        scanf("%d", &a[i]);
    }
    printf("Mang vua nhap la : ");
    for(int i=0;i<n;i++)
    {
        printf("%5d", a[i]);
    }
    int Dem = 0;
    printf("\nDanh sach cac so chinh phuong co trong mang\nvua nhap : ");
    for(int i=0;i<n;i++)
    {
        int b = (int)sqrt(a[i]);
        if(b*b == a[i])
        {
```

```
        printf("%5d", a[i]);  
        Dem++;  
    }  
    }  
    printf("\nSo luong so chinh phuong co trong mang vua  
nhap : %d", Dem);  
}
```

Bài 39.

Nhập vào từ bàn phím mảng số nguyên gồm n phần tử. Đếm và hiển thị các số nguyên tố có trong mảng.

* Thuật toán:

+ Duyệt từ đầu đến cuối mảng

+ Tại mỗi vị trí, nếu giá trị tại vị trí đang xét là số nguyên tố, tăng biến Dem lên 1, hiển thị giá trị lên màn hình. (Thực hiện kiểm tra SNT bằng cách đếm số ước của giá trị đó).

Ngược lại, chuyển đến vị trí tiếp theo.

* Chương trình:

```
#include <conio.h>  
#include <stdio.h>  
int main()  
{  
    int a[50], n;  
    printf("Nhap so luong phan tu : ");  
    scanf("%d", &n);  
    for(int i=0;i<n;i++)  
    {    printf("a[%d] = ", i);  
        scanf("%d", &a[i]);  
    }  
    printf("Mang vua nhap la : ");  
    for(int i=0;i<n;i++)  
    {  
        printf("%5d", a[i]);  
    }  
    int Dem = 0;  
    printf("\nDanh sach cac so nguyen to : ");  
    for(int i=0;i<n;i++)  
    {  
        if(a[i] > 0)  
        {  
            int SL_Uoc = 0;  
            for(int j=2;j<a[i];j++)
```

```
        {
            if(a[i]%j == 0)
            {
                SL_Uoc++;
            }
        }
        if(SL_Uoc == 0)
        {
            Dem++;
            printf("%5d", a[i]);
        }
    }
}
printf("\nSo luong so nguyen to : %d", Dem);
}
```

Bài 40.

Nhập vào từ bàn phím mảng số nguyên gồm n phần tử. Thay thế các phần tử âm bằng 0, hiển thị lên màn hình.

* Thuật toán:

- + Duyệt từ đầu đến cuối mảng
 - + Tại mỗi vị trí, nếu giá trị <0, thay thế giá trị bằng 0.
- Ngược lại, chuyển đến vị trí kế tiếp.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int a[50], n;
    printf("Nhap so luong phan tu : ");
    scanf("%d", &n);
    for(int i=0;i<n;i++)
    {
        printf("a[%d] = ", i);
        scanf("%d", &a[i]);
    }
    printf("Mang vua nhap la : ");
    for(int i=0;i<n;i++)
    {
        printf("%5d", a[i]);
    }
    for(int i=0;i<n;i++)
    {

```

```
        if(a[i]<0)
        {
            a[i] = 0;
        }
    }
    printf("Mang sau thay doi la : ");
    for(int i=0;i<n;i++)
    {
        printf("%5d", a[i]);
    }
}
```

Bài 41.

Nhập vào từ bàn phím mảng số nguyên gồm n phần tử. Nhập vào từ bàn phím số nguyên x, kiểm tra x có xuất hiện trong mảng hay không?

* Thuật toán:

- + Khởi tạo cờ KT = false (không tồn tại).
 - + Duyệt từ đầu đến cuối mảng
 - + Tại mỗi vị trí, nếu giá trị tại vị trí đang xét bằng x, thì chuyển trạng thái cờ KT = true, thoát vòng lặp (break).
- Ngược lại, chuyển đến vị trí tiếp theo.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int a[50], n;
    printf("Nhap so luong phan tu : ");
    scanf("%d", &n);
    for(int i=0;i<n;i++)
    {
        printf("a[%d] = ", i);
        scanf("%d", &a[i]);
    }
    printf("Mang vua nhap la : ");
    for(int i=0;i<n;i++)
    {
        printf("%5d", a[i]);
    }
    int x;
    printf("\Nhap x : ");
    scanf("%d",&x);
```

```
bool kt = false;
for(int i=0;i<n;i++)
{
    if(a[i] == x)
    {
        kt = true;
        break;
    }
}
if(kt == true)
{
    printf("%d có tồn tại trong mảng", x);
}
else
{
    printf("%d không tồn tại trong mảng", x);
}
}
```

Bài 42.

Nhập vào từ bàn phím mảng số nguyên gồm n phần tử. Kiểm tra xem mảng vừa nhập có tăng dần hay không?

* Thuật toán:

- + Khởi tạo cờ KT = true (mảng tăng dần).
- + Duyệt từ vị trí 1 đến n-1
- + Tại mỗi vị trí, nếu giá trị tại vị trí đang xét nhỏ hơn giá trị tại vị trí kề trước, thì chuyển trạng thái cờ KT = false, thoát vòng lặp (break).

Ngược lại, chuyển đến vị trí tiếp theo.

- + Kết thúc vòng lặp, nếu cờ KT = true, hiển thị dãy sắp xếp tăng dần.

Ngược lại, thông báo dãy không sắp xếp tăng dần.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int a[50], n;
    printf("Nhap so luong phan tu : ");
    scanf("%d", &n);
    for(int i=0;i<n;i++)
    {
```

```
        printf("a[%d] = ", i);
        scanf("%d", &a[i]);
    }
    printf("Mang vua nhap la : ");
    for(int i=0;i<n;i++)
    {
        printf("%5d", a[i]);
    }
    bool KT = true;
    for(int i=1;i<n;i++)
    {
        if(a[i] < a[i-1])
        {
            KT = false;
            break;
        }
    }
    if(KT == true)
    {
        printf("\nDay sap xep tang dan!");
    }
    else
    {
        printf("\nDay khong sap xep tang dan!");
    }
}
```

Bài 43.

Nhập vào từ bàn phím mảng số nguyên gồm n phần tử. Kiểm tra xem mảng vừa nhập có phải dãy đối xứng hay không?

* Thuật toán:

- + Khởi tạo cờ KT = true (mảng đối xứng).
- + Duyệt từ vị trí 0 đến n/2 (đúng với cả trường hợp n lẻ hay chẵn)
- + Tại mỗi vị trí, nếu giá trị tại vị trí đang xét khác giá trị đối xứng, thì chuyển trạng thái cờ KT = false, thoát vòng lặp (break).

Ngược lại, chuyển đến vị trí tiếp theo.

- + Kết thúc vòng lặp, nếu cờ KT = true, hiển thị dãy đối xứng.

Ngược lại, thông báo dãy không đối xứng.

* Chương trình:

```
#include <conio.h>
```

```
#include <stdio.h>
int main()
{
    int a[50], n;
    printf("Nhap so luong phan tu : ");
    scanf("%d", &n);
    for(int i=0;i<n;i++)
    {
        printf("a[%d] = ", i);
        scanf("%d", &a[i]);
    }
    printf("Mang vua nhap la : ");
    for(int i=0;i<n;i++)
    {
        printf("%5d", a[i]);
    }
    bool KT = true;
    for(int i=1;i<n/2;i++)
    {
        if(a[i] != a[n-i-1])
        {
            KT = false;
            break;
        }
    }
    if(KT == true)
    {
        printf("\nDay doi xung!");
    }
    else
    {
        printf("\nDay khong doi xung!");
    }
}
```

PHẦN 3: MA TRẬN

Bài 44.

Nhập vào ma trận A ($m \times n$), hiển thị ma trận A và ma trận chuyển vị của ma trận A lên màn hình.

* Thuật toán:

- + Ma trận chuyển vị đổi vị trí hàng và cột
- + Duyệt từ cột 0 đến cột $n-1$
 - + Trên mỗi cột, duyệt từ hàng 0 đến hàng $m-1$
 - + Tại mỗi vị trí xét, hiển thị giá trị của ma trận lên màn hình
 - + Hiển thị một dấu xuống dòng.

* Chương trình:

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

int main()
{
    int a[100][100], m, n;
    printf("Nhap m, n : ");
    scanf("%d%d", &m, &n);
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
        {
            printf("a[%d][%d] = ", i, j);
            scanf("%d", &a[i][j]);
        }
    }
    printf("Ma tran da nhap : \n");
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
        {
            printf("%5d", a[i][j]);
        }
        printf("\n");
    }
    printf("\nMa tran chuyen vi : \n");
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<m;j++)
```



```
        {  
            printf("%5d", a[j][i]);  
        }  
        printf("\n");  
    }  
}
```

Bài 45.

Nhập vào ma trận A (m x n), và ma trận B (m x n). Tính tổng $C = A + B$, hiển thị giá trị của C lên màn hình.

* Thuật toán:

+ Cộng tương ứng giữa hàng và hàng, cột và cột của ma trận.

* Chương trình:

```
#include <conio.h>  
#include <stdio.h>  
  
int main()  
{  
    int a[100][100], b[100][100], c[100][100], m, n;  
    printf("Nhap m, n : ");  
    scanf("%d%d", &m, &n);  
    printf("Nhap ma tran A : \n");  
    for(int i=0;i<m;i++)  
    {  
        for(int j=0;j<n;j++)  
        {  
            printf("a[%d][%d] = ", i, j);  
            scanf("%d", &a[i][j]);  
        }  
    }  
    printf("Nhap ma tran B : \n");  
    for(int i=0;i<m;i++)  
    {  
        for(int j=0;j<n;j++)  
        {  
            printf("b[%d][%d] = ", i, j);  
            scanf("%d", &b[i][j]);  
        }  
    }  
    for(int i=0;i<m;i++)  
    {  
        for(int j=0;j<n;j++)  
        {  
            c[i][j] = a[i][j] + b[i][j];  
        }  
    }  
}
```

```
    }  
}  
printf("\nMa tran C : \n");  
for(int i=0;i<m;i++)  
{  
    for(int j=0;j<n;j++)  
    {  
        printf("%5d", c[i][j]);  
    }  
    printf("\n");  
}  
}
```

Bài 46.

Nhập vào ma trận A (m x n), hiển thị ma trận A màn hình. Tính tổng các phần tử có trong ma trận

* Thuật toán:

- + Duyệt i từ hàng 0 đến hàng m
- + Trên mỗi hàng, duyệt j từ cột 0 đến cột n
- + Tong=Tong+a[i][j]
- + Hiển thị Tong

* Chương trình:

```
#include <conio.h>  
#include <stdio.h>  
  
int main()  
{  
    int a[100][100], m, n;  
    printf("Nhap m, n : ");  
    scanf("%d%d", &m, &n);  
    for(int i=0;i<m;i++)  
    {  
        for(int j=0;j<n;j++)  
        {  
            printf("a[%d][%d] = ", i, j);  
            scanf("%d", &a[i][j]);  
        }  
    }  
    printf("Ma tran da nhap : \n");  
    for(int i=0;i<m;i++)  
    {
```

```
        for(int j=0;j<n;j++)
        {
            printf("%5d", a[i][j]);
        }
        printf("\n");
    }
    int Tong=0;
    for(int i=0;i<m;i++)
        for(int j=0;j<n;j++)
            Tong=Tong+a[i][j];
    printf("\nTong cac phan tu cua ma tran =%d",Tong);
}
```

Bài 47.

Nhập vào ma trận A (m x n), hiển thị ma trận A màn hình. Tính tổng các phần tử trên hàng 0 của ma trận

* Thuật toán:

- +Trên hàng 0
- + Duyệt j từ cột 0 đến cột n
- + Tong=Tong+a[0][j]
- + Hiển thị Tong

* Chương trình:

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

int main()
{
    int a[100][100], m, n;
    printf("Nhap m, n : ");
    scanf("%d%d", &m, &n);
    for(int i=0;i<m;i++)
    {
        for(int j=0;j<n;j++)
        {
            printf("a[%d][%d] = ", i, j);
            scanf("%d", &a[i][j]);
        }
    }
    printf("Ma tran da nhap : \n");
    for(int i=0;i<m;i++)
    {
```

```
        for(int j=0;j<n;j++)
        {
            printf("%5d", a[i][j]);
        }
        printf("\n");
    }
    int Tong=0;
    for(int j=0;j<n;j++)
        Tong=Tong+a[0][j];
    printf("\nTong cac phan tu cua hang 0 =%d",Tong);
}
```

Bài 48.

Nhập vào ma trận A ($n \times n$), hiển thị ma trận A theo dạng bảng. Tính tổng đường chéo chính của ma trận.

* Thuật toán:

- + Đường chéo chính của ma trận có giá trị hàng bằng giá trị cột =>
- + Duyệt từ 1 đến n (duyệt qua các hàng của ma trận).
- + Cộng dồn giá trị Tong với giá trị tại ô có giá trị hàng và cột trùng nhau.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int a[100][100], n;
    printf("Nhap n : ");
    scanf("%d", &n);
    printf("Nhap ma tran A : \n");
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            printf("a[%d][%d] = ", i, j);
            scanf("%d", &a[i][j]);
        }
    }
    printf("\nMa tran A vua nhap : \n");
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
```

```
        printf("%5d", a[i][j]);  
    }  
    printf("\n");  
}  
int Tong = 0;  
for(int i=0;i<n;i++)  
{  
    Tong += a[i][i];  
}  
printf("Tong duong cheo chinh : %d", Tong);  
}
```

Bài 49.

Nhập vào ma trận A ($n \times n$), hiển thị ma trận A theo dạng bảng. Hiển thị các số thuộc đường chéo phụ của ma trận a ra màn hình.

* Thuật toán:

- + Đường chéo chính của ma trận có giá trị cột bằng ($n - \text{số hàng} - 1$) =>
- + Duyệt từ 1 đến n (duyệt qua các hàng của ma trận).
- + Hiển thị giá trị tại ô có chỉ số cột bằng $n - \text{chỉ số hàng} - 1$.

* Chương trình:

```
#include <conio.h>  
#include <stdio.h>  
  
int main()  
{  
    int a[100][100], n;  
    printf("Nhap n : ");  
    scanf("%d", &n);  
    printf("Nhap ma tran A : \n");  
    for(int i=0;i<n;i++)  
    {  
        for(int j=0;j<n;j++)  
        {  
            printf("a[%d][%d] = ", i, j);  
            scanf("%d", &a[i][j]);  
        }  
    }  
    printf("\nMa tran A vua nhap : \n");  
    for(int i=0;i<n;i++)  
    {  
        for(int j=0;j<n;j++)  
        {  
            printf("%5d", a[i][j]);  
        }  
    }  
}
```

```
    }  
    printf("\n");  
}  
printf("\nCac phan tu thuoc duong cheo phu : ");  
for(int i=0;i<n;i++)  
{  
    printf("%5d", a[i][n-i-1]);  
}  
}
```

Bài 50.

Nhập vào ma trận A (m x n), hiển thị ma trận A theo dạng bảng. Tìm số lớn nhất trên hàng thứ 2 của ma trận.

* Thuật toán:

+ Nếu số lượng hàng < 3, thông báo không có hàng thứ 2 của ma trận

Ngược lại, gán Max = giá trị tại hàng 2 cột 0.

Duyệt từ cột 1 đến cột n-1, xét giá trị tại hàng 2 và cột tương ứng nếu lớn hơn Max thì cập nhật lại giá trị của Max, ngược lại chuyển đến vị trí cột kế tiếp.

* Chương trình:

```
#include <conio.h>  
#include <stdio.h>  
  
int main()  
{  
    int a[100][100], m, n;  
    printf("Nhap n : ");  
    scanf("%d%d", &m, &n);  
    printf("Nhap ma tran A : \n");  
    for(int i=0;i<m;i++)  
    {  
        for(int j=0;j<n;j++)  
        {  
            printf("a[%d][%d] = ", i, j);  
            scanf("%d", &a[i][j]);  
        }  
    }  
    printf("\nMa tran A vua nhap : \n");  
    for(int i=0;i<m;i++)  
    {  
        for(int j=0;j<n;j++)  
        {  
            printf("%5d", a[i][j]);  
        }  
    }  
}
```

```
    }
    printf("\n");
}
if(m < 3)
{
    printf("Ma tran khong co cot 2!");
}
else
{
    int Max = a[2][0];
    for(int i=1;i<n;i++)
    {
        if(a[2][i] > Max)
        {
            Max = a[2][i];
        }
    }
    printf("Max tai cot 2 la : %d", Max);
}
}
```

Bài 51.

Nhập vào ma trận A ($n \times n$), hiển thị ma trận A theo dạng bảng. Đếm các số chẵn có trong mảng.

* Thuật toán:

- + Duyệt từ hàng 0 đến hàng $n-1$ của ma trận
- + Duyệt từ cột 0 đến cột $n-1$ của ma trận
- + Tại mỗi vị trí, nếu giá trị là chẵn (chia hết cho 2) thì tăng biến

Dem lên 1 đơn vị.

Ngược lại, chuyển qua vị trí kế tiếp.

* Chương trình:

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

int main()
{
    int a[100][100], n;
    printf("Nhap n : ");
    scanf("%d", &n);
    printf("Nhap ma tran A : \n");
    for(int i=0;i<n;i++)
    {
```

```
        for(int j=0;j<n;j++)
        {
            printf("a[%d][%d] = ", i, j);
            scanf("%d", &a[i][j]);
        }
    }
    printf("\nMa tran A vua nhap : \n");
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            printf("%5d", a[i][j]);
        }
        printf("\n");
    }
    int Dem = 0;
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            if(a[i][j]%2==0)
            {
                Dem++;
            }
        }
    }
    printf("So luong so chan trong ma tran : %d", Dem);
}
```


PHẦN 4: XÂU KÝ TỰ

Bài 52.

Nhập vào từ bàn phím chuỗi s, hiển thị chuỗi s và chuỗi đảo của s lên màn hình.

* Thuật toán:

+ Nhập vào chuỗi s từ bàn phím

+ Chuỗi đảo của chuỗi s:

Duyệt từ cuối chuỗi về đầu chuỗi (strlen(s) để lấy độ dài chuỗi), tại mỗi vị trí, hiển thị giá trị ký tự tại vị trí đó trong chuỗi s.

* Chương trình:

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

int main()
{
    char s[100];
    printf("Nhap s = ");
    fflush(stdin);
    gets(s);
    printf("\nXau s la : %s", s);
    printf("\nXau dao cua s la : ");
    for(int i=strlen(s)-1; i>=0; i--)
    {
        printf("%c", s[i]);
    }
}
```

Bài 53.

Nhập vào từ bàn phím chuỗi s, hiển thị chuỗi s lên màn hình, đếm số ký tự 'a' xuất hiện trong chuỗi.

* Thuật toán:

+ Nhập vào chuỗi s từ bàn phím

+ Duyệt từ đầu đến cuối chuỗi (strlen(s) để lấy độ dài chuỗi)

+ Tại mỗi vị trí, nếu ký tự đang xét là 'a' thì tăng biến Dem lên 1 đơn vị,

Ngược lại, chuyển đến vị trí kế tiếp.

* Chương trình:

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

int main()
{
    char s[100];
    printf("Nhap s = ");
    fflush(stdin);
    gets(s);
    printf("Xau s la : %s", s);
    int Dem=0;
    for(int i=0;i<strlen(s);i++)
    {
        if(s[i] == 'a')
        {
            Dem++;
        }
    }
    printf("\nSo luong ky tu 'a' la : %d", Dem);
}
```

Bài 54.

Nhập vào từ bàn phím xâu s, hiển thị xâu s lên màn hình. Giả sử, xâu s đã được chuẩn hóa (không có khoảng trống thừa), đếm số từ có trong xâu.

* Thuật toán:

- + Nhập vào xâu s từ bàn phím
- + Nếu độ dài xâu = 0 (xâu rỗng), thông báo xâu rỗng.

Ngược lại, đếm số lượng ký tự cách có trong xâu (do mỗi từ được cách nhau bởi 1 khoảng trống). Số từ = số ký tự cách + 1.

* Chương trình:

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

int main()
{
    char s[100];
    printf("Nhap s = ");
    fflush(stdin);
    gets(s);
    printf("Xau s la : %s", s);
}
```

```
if(strlen(s)==0)
{
    printf("\nXau la xau rong!");
}
else
{
    int Dem=0;
    for(int i=0;i<strlen(s);i++)
    {
        if(s[i] == ' ')
        {
            Dem++;
        }
    }
    printf("\nSo luong tu trong xau la : %d", Dem+1);
}
```

Bài 55.

Nhập vào từ bàn phím xâu s, hiển thị xâu s lên màn hình. Giả sử, xâu s đã được chuẩn hóa (không có khoảng trống thừa), hãy chuyển đổi chuỗi ký tự thành viết hoa các ký tự đầu tiên của mỗi từ.

* Thuật toán:

+ Nhập vào xâu s từ bàn phím

+ Nếu độ dài xâu = 0 (xâu rỗng), thông báo xâu rỗng.

Ngược lại, duyệt từ đầu đến cuối xâu

+ Tại mỗi vị trí, nếu (vị trí đang xét là vị trí đầu tiên hoặc liền trước vị trí đang xét là một ký tự cách, chuyển ký tự về dạng in hoa).

Ngược lại, chuyển đến vị trí kế tiếp.

* Chương trình:

```
#include <conio.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>
int main()
{
    char s[100];
    printf("Nhap s = ");
    fflush(stdin);
    gets(s);
    printf("Xau s la : %s", s);
```

```
if(strlen(s)==0)
{
    printf("\nXau la xau rong!");
}
else
{
    for(int i=0;i<strlen(s);i++)
    {
        if(i==0 || s[i-1] == ' ')
        {
            s[i] = toupper(s[i]);
        }
    }
}
printf("\nXau s sau khi chuyen doi : %s", s);
}
```