



BÀI THỰC HÀNH SỐ 1

Nhập/xuất, lệnh điều kiện, toán tử ?

1. BÀI TẬP 1

Nhập vào một số nguyên n ($32 \leq n \leq 255$) và in ra ký tự có mã ASCII của n .

Chương trình:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int n; /*khai bao bien n*/
    printf("Nhap n = "); /*hien thi dong thong bao nhap n*/
    scanf("%d", &n); /*nhap n tu ban phim*/
    printf("Ky tu co ma ASCII cua n: %c", n);
}
```

2. BÀI TẬP 2

Viết chương trình nhập giờ h , phút m và giây s hiện tại và một số nguyên dương n . In ra màn hình giờ, phút, giây sau n giây.

Thuật toán:

Bước 1: Nhập giờ h , phút m , giây s và số tự nhiên n ;

Bước 2: Tính giờ phút giây sau n giây theo công thức dưới đây;

$$\begin{aligned}h &= (h + ((n + s)/60 + m)/60) \% 24; \\m &= (m + (n + s)/60) \% 60; \\s &= (n + s) \% 60;\end{aligned}$$

Bước 3: In ra giá trị giờ, phút và giây mới.

Chương trình:

```
#include <stdio.h>
int main()
{
    int n, h, m, s;
    printf("Nhap gio: ");
    scanf("%d", &h);
    printf("Nhap phut: ");
    scanf("%d", &m);
    printf("Nhap giay: ");
    scanf("%d", &s);
    printf("Nhap so nguyen n: ");
```



```
scanf("%d", &n);
h = (h + ((n + s) / 60 + m) / 60) % 24;
m = (m + (n + s) / 60) % 60;
s = (n + s) % 60;
printf("Ket qua gio, phut, giay sau n giay: ");
printf("%d : %d : %d\n", h, m, s);
}
```

3. BÀI TẬP 3

Viết chương trình nhập bán kính R và xuất ra màn hình diện tích S và chu vi L của hình tròn (diện tích $S = \pi R^2$ và chu vi $L = 2\pi R$).

Thuật toán:

Bước 1: Nhập bán kính R;

Bước 2: Tính diện tích $S = \pi R^2$ và $L = 2\pi R$. Sử dụng hằng số π (M_PI) thuộc thư viện math.h;

Bước 3: In ra giá trị của diện tích và chu vi.

Chương trình:

```
#include <stdio.h>
#include <math.h>
int main()
{
    float R, S, L; /*Khai bao cac bien*/
    printf("Nhap ban kinh R = "); /*Thong bao nhap ban kinh r*/
    scanf("%f", &R); /*Nhap ban kinh R tu ban phim*/
    S = R*R*M_PI; /*M_PI la so PI*/
    L = 2*M_PI*R;
    printf("Dien tich hinh tron = %0.3f\n", S);
    printf("Chu vi hinh tron = %0.3f\n", L);
}
```

4. BÀI TẬP 4

Viết chương trình nhập số thực x và tính giá trị các hàm một biến sau:

$$f(x) = \frac{\sqrt[3]{x^2} + x^5 + \ln(x^4 + 1)}{x^2 + \frac{\pi}{4}}$$

Thuật toán:

Bước 1: Nhập vào số thực x;

Bước 2: Tính tử số của biểu thức lưu vào biến tuso. Sử dụng hàm pow() để tính lũy thừa và hàm log() để tính logarit, các hàm này thuộc thư viện math.h;

Bước 3: Tính mẫu số của biểu thức lưu vào biến mauso;



Bước 4: Tính hàm $f(x) = \text{tuso/mauso}$ và in ra kết quả.

Chương trình:

```
#include <stdio.h>
#include <math.h>
int main()
{
    float ketqua, tu, mau;
    float x;
    printf("Nhap x: ");
    scanf("%f", &x);
    tu = pow(x*x, 1.0/3) + pow(x, 5) + log(pow(x, 4)+1);
    mau = x*x + M_PI/4;
    ketqua = tu/mau;
    printf("Gia tri cua bieu thuc = %4.2f", ketqua);
    return 0;
}
```

1. BÀI TẬP 5

Viết chương trình nhập 03 số a,b,c từ bàn phím. Tìm số trung gian của 03 số này bằng toán tử ?.

Ví dụ : có 03 số: 3,4,5 thì số trung gian là số 4.

Thuật toán:

Bước 1: Nhập vào số a,b,c

Bước 2: Tìm giá trị lớn nhất, nhỏ nhất của 3 số đó

Bước 3: Lấy tổng của 3 số trừ đi cho giá trị lớn nhất và nhỏ nhất

Bước 4: In ra kết quả.

Chương trình:

```
#include <stdio.h>
main()
{
    float a,b,c,max,min;
    printf("nhap a,b,c:");
    scanf("%f%f%f", &a, &b, &c);
    max=a>b&&a>c?a:(b>c?b:c);
    min=a<b&&a<c?a:(b<c?b:c);
    printf("so can tim la:%f",a+b+c-max-min);
}
```



2. BÀI TẬP 6

Viết chương trình nhập một chữ cái ch và in ra chữ hoa nếu ch là chữ thường; ngược lại, in ra chữ thường nếu ch là chữ hoa.

Nhận xét: Chúng ta nhận thấy rằng mã ASCII của chữ cái thường bằng mã ASCII của chữ cái hoa cộng 32. Chẳng hạn, chữ cái 'a' có mã ASCII là 97 thì chữ cái 'A' có mã ASCII là 65.

Thuật toán:

Bước 1: Nhập vào chữ cái ch.

Bước 2: Nếu là chữ cái thường, nghĩa là $ch \geq 'a'$ và $ch \leq 'z'$ thì chuyển thành chữ cái hoa, nghĩa là $ch = ch - 32$.

Bước 3: Nếu là chữ cái hoa, nghĩa là $ch \geq 'A'$ và $ch \leq 'Z'$ thì chuyển thành chữ cái thường, nghĩa là $ch = ch + 32$.

Chương trình:

```
#include<stdio.h>
int main()
{
    char ch;
    printf("Nhap vao mot chu cai : ");
    scanf("%c", &ch);
    if (ch>='a' && ch<='z')
        printf("Chuyen thanh chu cai hoa: %c\n", ch -= 32);
    else if (ch>='A' && ch<='Z')
        printf("Chuyen thanh chu cai thuong: %c\n", ch += 32);
    return 0;
}
```

3. BÀI TẬP 7

Viết chương trình nhập hai số thực x, y và tính giá trị hàm sau:

$$f(x, y) = \frac{\sqrt[3]{x^2 + y^2} - \log_5(x + y)}{\arctg(x - y) + \frac{\pi}{4}}$$

Thuật toán:

Bước 1: Nhập vào hai số thực x và y;

Bước 2: Nếu $(x + y) > 0$ và nếu mẫu số khác không thì tính hàm f, và in ra kết quả. Sử dụng các hàm pow(), log() và atan() trong thư viện math.h;

Bước 3: Ngược lại, nếu $(x + y) \leq 0$ thì không tính được $\log_5(x + y)$, nên báo lỗi.

Chương trình:

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



```
float x, y, tuso, mauso, f;
printf("Nhap x = ");
scanf("%f", &x);
printf("Nhap y = ");
scanf("%f", &y);
if ((x + y) > 0)
{
    mauso = atan(x-y) + M_PI/4;
    if(mauso == 0) printf("Loi mau so bang 0!\n");
    else
    {
        tuso = pow(x*x + y*y, 1.0/3) - log(x+y)/log(5);
        f = tuso / mauso;
        printf("Gia tri cua ham=%4.2f", f);
    }
}
else
printf("Loi mien xac dinh cua ham logarit!\n");
}
```

4. BÀI TẬP 8

Viết chương trình giải phương trình bậc 2 ($ax^2 + bx + c = 0$, $a \neq 0$)

Thuật toán:

Bước 1: Nhập các hệ số a, b và c;

Bước 2: Tính $\Delta = b^2 - 4ac$;

Bước 3: Nếu $\Delta > 0$ thì tính và in hai nghiệm:

$$x_1 = \frac{-b - \sqrt{\Delta}}{2a};$$

$$x_2 = \frac{-b + \sqrt{\Delta}}{2a};$$

Bước 4: Nếu ngược lại ($\Delta \leq 0$), nếu $\Delta = 0$ thì tính và in nghiệm kép $x = -b/(2a)$;

Bước 5: Ngược lại nếu ($\Delta < 0$), thì in ra phương trình vô nghiệm.

Chương trình:

```
#include <stdio.h>
#include <math.h>
int main()
{
    float a, b, c, delta, x1, x2;
    printf("Giai phuong trinh bac hai\n");
    printf("Nhap he so a = ");
    scanf("%f", &a);
    printf("Nhap he so b = ");
    scanf("%f", &b);
    printf("Nhap he so c = ");
    scanf("%f", &c);
    delta = b*b - 4*a*c;
```



```
if (delta > 0)
{
    printf("Phuong trinh co hai nghiệm phân biệt\n");
    x1 = (-b - sqrt(delta))/(2*a);
    x2 = (-b + sqrt(delta))/(2*a);
    printf("Nghiệm x1 = %4.2f\n", x1);
    printf("Nghiệm x2 = %4.2f\n", x2);
}
else
if (delta == 0) printf("Phuong trinh co nghiệm kép
                    la: %4.2f\n", -b/(2*a));
else printf ("Phuong trinh vo nghiệm.\n");
}
```

Nhận xét: Có thể cải tiến chương trình trên để tính nghiệm cả trong trường hợp hệ số a bằng 0.

5. BÀI TẬP 9

Viết chương trình mô phỏng cách gọi menu. Chẳng hạn, nếu nhập vào chữ f hay F thì in ra là bạn chọn menu *File*; nếu nhập vào chữ h hay H thì in ra là bạn chọn menu *Help*.

Thuật toán:

Bước 1: Nhập vào ký tự ch;

Bước 2: Kiểm tra giá trị của ch;

- Nếu ch = 'F' thì in ra "Ban chon menu File"
- Nếu ch = 'f' thì in ra "Ban chon menu File"
- Nếu ch = 'H' thì in ra "Ban chon menu Help"
- Nếu ch = 'h' thì in ra "Ban chon menu Help"

Chương trình:

```
#include <stdio.h>
#include <math.h>
int main()
{
    char ch;
    printf("Nhap vao mot ky tu: ");
    scanf("%c", &ch);
    switch (ch)
    {
        case 'F':
        case 'f': printf("Ban chon menu File"); break;
        case 'H':
        case 'h': printf("Ban chon menu Help"); break;
        default : printf("Ban da nhap sai ky tu yeu cau");
    }
}
```



}

6. BÀI TẬP TỰ GIẢI

Bài 1. Viết chương trình nhập bán kính R và xuất ra màn hình thể tích V , diện tích S của hình cầu ($V = 4\pi R^3/3$; $S = 4\pi R^2$).

Bài 2. Viết chương trình nhập chiều dài L và chiều rộng W và xuất ra màn hình diện tích D và chu vi C của hình chữ nhật ($D = L \times W$ và $C = 2(L+W)$).

Bài 3. Viết chương trình nhập số thực x và tính giá trị của hàm một biến sau:

$$g(x) = \frac{x^3 + \log_5(|x-3|+2)}{\arctg^2(x-5)+1}$$

Bài 4. Viết chương trình nhập hai số thực x, y và tính giá trị của hàm hai biến sau:

$$g(x, y) = \frac{e^{x+y} + \ln(|x^2 - y| + 5)}{\cos(x+y) - \sin(x) + 3}$$

Bài 5. Viết chương trình nhập đáy trên a , đáy dưới b và chiều cao h của một hình thang. Tính diện tích của hình thang này.

Bài 6. Viết chương trình nhập hai số thực x, y và tính giá trị hàm sau:

$$g(x, y) = \frac{e^{x+y} - \ln(x^2 - y)}{\cos(x+y) - \sin(x)}$$

Lưu ý kiểm tra lỗi khi hàm không xác định.

Bài 7. Viết chương trình giải phương trình bậc bốn trùng phương:

$$(ax^4 + bx^2 + c = 0, a \neq 0)$$

Lưu ý đếm số nghiệm khác nhau của phương trình.

Bài 8. Viết chương trình giải hệ phương trình gồm hai phương trình và hai ẩn x, y với các hệ số nhập từ bàn phím.



BÀI THỰC HÀNH SỐ 2

Lệnh vòng lặp và hàm

1. BÀI TẬP 1

Viết chương in ra bảng cửu chương có dạng như sau:

2x1=2	3x1=3	4x1=4	5x1=5	6x1=6	7x1=7	8x1=8	9x1=9	10x1=10
2x2=4	3x2=6	4x2=8	5x2=10	6x2=12	7x2=14	8x2=16	9x2=18	10x2=20
2x3=6	3x3=9	4x3=12	5x3=15	6x3=18	7x3=21	8x3=24	9x3=27	10x3=30
2x4=8	3x4=12	4x4=16	5x4=20	6x4=24	7x4=28	8x4=32	9x4=36	10x4=40
2x5=10	3x5=15	4x5=20	5x5=25	6x5=30	7x5=35	8x5=40	9x5=45	10x5=50
2x6=12	3x6=18	4x6=24	5x6=30	6x6=36	7x6=42	8x6=48	9x6=54	10x6=60
2x7=14	3x7=21	4x7=28	5x7=35	6x7=42	7x7=49	8x7=56	9x7=63	10x7=70
2x8=16	3x8=24	4x8=32	5x8=40	6x8=48	7x8=56	8x8=64	9x8=72	10x8=80
2x9=18	3x9=27	4x9=36	5x9=45	6x9=54	7x9=63	8x9=72	9x9=81	10x9=90
2x10=20	3x10=30	4x10=40	5x10=50	6x10=60	7x10=70	8x10=80	9x10=90	10x10=100

Thuật toán:

Cho i=1 đến 10

Cho j=2 đến 10 sau đó in i*j.

Chương trình:

```
#include<stdio.h>
int main()
{
    printf("_____Bang cuu chuong_____\n\n");
    int i,j;
    for(i=1;i<=10;i++)
    {
        for(j=2;j<=10;j++)

            printf("%dx%d=%2d ",j,i,i*j);
        printf("\n");
    }
}
```




2. BÀI TẬP 2

Công thức đổi độ Fahrenheit F ra độ Celsius C là: $C = 5(F - 32) / 9$. Viết chương trình in ra bảng chuyển đổi với $0 \leq F \leq 300$ với bước nhảy 20. Dùng giá trị nguyên cho F và giá trị thực cho C.

Thuật toán:

Bước 1: Cho F = 0;

Bước 2: Nếu F > 300 thì dừng;

Bước 3: Tính $C = 5(F - 32) / 9$. In F và C trên 1 dòng;

Bước 4: Cho F = F + 20. Quay lại bước 2.

Chương trình:

```
#include <stdio.h>
int main()
{
    int F;
    float C;
    for (F=0; F<=300; F+=20)
    {
        C=5*(F-32)/9.0;
        printf("\n%-5d F <--> %.2f", F, C);
    }
}
```

3. BÀI TẬP 3

Giả sử lãi suất hằng tháng là $d=2\%$, tiền vay ngân hàng là $T = 1000000$. Viết chương trình nhập số nguyên dương n và in ra số tiền phải trả sau n tháng.

Thuật toán:

Bước 1: Nhập n;

Bước 2: Cho T = 1000000;

Bước 3: Cho i chạy từ 0 đến n - 1 tính: $T = 1.02T$;

Bước 4: In ra T.

Giải thích:

Gọi T_k là số tiền phải trả sau k tháng. Số tiền này bằng số tiền phải trả được tính cho đến tháng trước (T_{k-1}) cộng tiền lãi 1 tháng cuối cùng ($2\% T_{k-1} = 0.02T_{k-1}$). Có công thức truy hồi cho T_k là: $T_k = T_{k-1} + 0.02T_{k-1} = 1.02T_{k-1}$.

Yêu cầu:

Chạy chương trình nhiều lần với việc thay T là giá trị nhập từ bàn phím.

Chương trình:

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



```
int k, n;
float T=1000000;
printf("n= "); scanf("%d",&n);
for (k=0; k<n; k++) T*=1.02;
printf("So tien phai tra = %0.2f", T);
}
```

4. BÀI TẬP 4

Viết chương trình nhập số nguyên dương n và tính tổng:

$$S_n = \frac{1}{1} + \frac{1}{2} + \dots + \frac{1}{n}$$

Thuật toán:

Bước 1: Nhập n;

Bước 2: Cho S = 0;

Bước 3: Cho k chạy từ 1 đến n tính:

$$S = S + \frac{1}{k}$$

Bước 4: In ra S.

Nhận xét:

Phép chia dùng cho cả số thực lẫn số nguyên. Ngôn ngữ C luôn ngầm định là phép chia kiểu số nguyên (int).

Chương trình:

```
#include <stdio.h>
int main()
{
    int k, n;
    float S=0;
    printf("n= "); scanf("%d",&n);
    for (k=1; k<=n; k++) S+=1.0/k;
    printf("Sn = %0.2f", S);
}
```

5. BÀI TẬP 5

Số hoàn hảo là số nguyên dương bằng tổng các ước thực sự của nó. Ví dụ: 6 = 1+2+3. Viết chương trình in ra tất cả các số hoàn hảo nhỏ hơn 1000.

Thuật toán:

Cho n chạy từ 1 đến 999 thực hiện các bước sau:

Bước 1: Cho S = 0, k = 1;



Bước 2: Lặp cho đến khi $k > \frac{n}{2}$: Nếu k là ước của n thì cộng k vào S ;

Bước 3: Nếu $S = n$ thì in ra n .

Yêu cầu:

Thay số 1000 trong chương trình bởi số lớn hơn để tìm thêm các số hoàn hảo.

Chương trình:

```
#include <stdio.h>
int main()
{
    int n, k, S;
    for (n=1; n<1000; n++){
        S=0;
        for (k=1; k<=n/2; k++){
            if (n%k==0) S+=k;
        }
        if (S==n) printf("\n%d", n);
    }
}
```

6. BÀI TẬP 6

Viết chương trình in ra tất cả các số hoàn hảo nhỏ hơn n có sử dụng hàm :

Chương trình:

```
#include <stdio.h>
#include <math.h>
int Hoanhao(int n)
{
    int i, sum;
    sum=0;
    for(i=1; i<=n/2; i++)
        if(n%i==0) sum+=i;
    if(sum==n) return 1 ;
    else return 0;
}
main()
{
    int n, i;
    printf("nhap so n ="); scanf("%d", &n);
    for(i=1; i<=n; i++)
        if (Hoanhao(i)==1) printf("%10d", i);
}
```

7. BÀI TẬP 7

Viết chương trình in ra tất cả các số nguyên tố từ 2 đến 1000000.

**Thuật toán:**

Thuật toán đơn giản kiểm tra n có phải là số nguyên tố hay không và in n nếu nguyên tố.

Bước 1: Cho $nt = 1$, $k = 2$;

Bước 2: Lặp trong khi $k < n$:

- Nếu k là ước của n thì $nt = 0$;
- Cho $k = k + 1$.

Bước 3: Nếu $nt = 1$ thì in ra n .

Nhận xét:

a) Ngôn ngữ C ngầm định hằng số kiểu nguyên (int). Nên với hằng số nguyên lớn (long int) thì thêm L ở cuối. Như 123456L.

b) Nếu $n = x.y$ và nếu $y > \sqrt{n}$ thì $x < \sqrt{n}$. Do đó chỉ cần kiểm tra đến $k = \sqrt{n}$ hay $k^2 = n$. Hơn nữa, nếu k là ước của n thì không cần kiểm tra tiếp.

Vậy có thể cải tiến cho chương trình chạy nhanh hơn bằng cách thay bước kiểm tra n nguyên tố như sau:

- Tăng k lên một đơn vị trong khi $k^2 \leq n$ và k không là ước của n .
- Nếu $k^2 > n$ thì n là số nguyên tố.

Yêu cầu:

Thay số 1000000L trong chương trình bởi số lớn hơn như 1000000000L để theo dõi thời gian chạy chương trình.

Chương trình:

```
#include <stdio.h>
int main()
{
    long n, k, nt;
    for (n=2; n<1000000L; n++){
        nt=1;
        for (k=2; k<n; k++)
            if (n%k==0) nt=0;
        if (nt) printf("\n%d", n);
    }
}
```

8. BÀI TẬP 8

Viết chương trình có sử dụng hàm kiểm tra n (nhập từ bàn phím) có phải số nguyên tố không ?



Chương trình:

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

int kt(int n)
{
    int i;
    if(n<2) return 0;
    for(i=2;i<=(int)sqrt(n); i++)
        if(n%i==0) return 0;
    return 1;
}

main()
{
    int n;
    printf("Nhap n=") ;
    scanf("%d",&n);
    if(kt(n)) printf("Day la so nguyen to"); else printf("Khong phai
la so nguyen to");
}
```

9. BÀI TẬP 9

Viết chương trình có sử dụng hàm in ra các nguyên tố nhỏ hơn hoặc bằng n (n nhập từ bàn phím) ?

Chương trình:

```
#include <stdio.h>
#include <math.h>
int kt(int n)
{
    int i;
    if(n<2) return 0;
    for(i=2;i<=(int)sqrt(n); i++)
        if(n%i==0) return 0;
    return 1;
}

main()
{
    int i,n;
    printf("Nhap n=") ;
    scanf("%d",&n);
    for (i=0; i<=n; i++)
        if(kt(i)) printf("%4d",i);
}
```

10. BÀI TẬP 10

Viết chương trình in ra màn hình bảng chân trị của các phép toán logic:



AND, OR, NAND, NOT, NOR, XOR. Dùng các phép logic tương ứng là: &&, ||, !, ^.

Cụ thể bảng chân trị như sau: Y là kết quả của các phép toán logic trên

Phép NOT:

A	Y
0	1
0	1
1	0
1	0

Phép AND

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Phép OR

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Phép NOR

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

Phép NAND

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

Phép XOR

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0



Chương trình:

```
#include <stdio.h>
#define TRUE 1
#define FALSE 0
/*--KHAI BAO CAC HAM NGUYEN MAU--*/
/*-----*/
int NOT(int x);
int AND(int x,int y);
int OR(int x,int y);
int NAND(int x,int y);
int NOR(int x,int y);
int XOR(int x,int y);
void VIET();
/*-----*/
int main()
{
char ch;
tt:
printf("\n\tCopyright by NGUYEN VAN NGUYEN \n");
printf("\n");
printf("\n\tBang chan tri cua dinh luat DE Morgan");
printf("\n");
VIET();
printf("\n");
printf("\nAn Enter tiep tục.An phim ESC de thoát");
do
{
ch=getch();
}while(ch!=13 && ch!=27);
if(ch==13) goto tt;
return(0);
}
/*-----*/
int AND(int x,int y)
{
return(x&y);
}
/*-----*/
int OR(int x,int y)
{
return(x||y);
}
/*-----*/
int NOR(int x,int y)
{
return(!(x||y));
}
```



```
/*-----*/
int NAND(int x,int y)
{
    return(!(x&y));
}
/*-----*/
int NOT(int x)
{
    return(!x);
}
int XOR(int x,int y)
{
    return (x^y);
}
/*-----*/
void VIET()
{
    int A,B,C,D,E,F,G,H;
    printf("\n\tA\tB\tC\tD\tE\tF\tG\tH");

    for (A=FALSE;A<=TRUE;A++)
        for (B=FALSE;B<=TRUE;B++)
        {
            C=AND (A, B) ;
            D=OR (A, B) ;
            E=NOR (A, B) ;
            F=NAND (A, B) ;
            G=NOT (A) ;
            H=XOR (A, B) ;
            printf("\n\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d",A,B,C,D,E,F,G,H) ;
        }
}
```

11. BÀI TẬP 11

Viết chương trình tính $N!!$, với N là số tự nhiên ≤ 19 và được nhập từ bàn phím.

N chẵn: $N!! = 2 * 4 * 6 * \dots * N$.

N lẻ: $N!! = 1 * 3 * 5 * \dots * N$

- Dùng for
- Dùng while
- Dùng do while
- Dùng đệ quy

Chương trình

```
#include <stdio.h>
#include <math.h>
long gtkep(long n)
```




```
{ long p=1;
  int i;
  for(i=n; i>0; i-=2) p*=i;
  return p;
}
long gtk(long n)
{
  long p=1;
  while (n>0) { p=p*n;
                n-=2;}
  return p;
}
int giaithua(int n)
{
  if(n<2) return 1;
  return n*giaithua(n-2);
}
long gt(long n)
{ long p=1;
  do { p*=n; n-=2;}
  while (n>0);
  return p;
}
main()
{int n;
printf("Nhap n=");
scanf("%ld",&n);
printf("Gt kep =%ld\n",gtkep(n));
printf("Gt kep =%ld\n",gtk(n));
printf("Gt kep =%ld\n",giaithua(n));
printf("gt kep = %ld\n",gt(n));
}
```

12 BÀI TẬP 12

các cách hoán đổi hai số nhập từ bàn phím

Chương trình



```
#include <stdio.h>
void Swap1(int &a, int &b) // Hoan doi khong dung bien tam
{
    a=a+b;
    b=a-b;
    a=a-b;
}
void Swap2(int &a, int &b) // Hoan doi khong dung bien tam. Dung phep
XOR
{
    a = a ^ b;
    b = a ^ b;
    a = a ^ b;
}

void Swap3(int &a, int &b) // Hoan doi dung bien tam
{
    int tam;
    tam=a;
    a=b;
    b=tam;
}
void Swap4(int *a, int *b) // Hoan doi dung con tro va bien tam
{
    int tam;
    tam=*a;
    *a=*b;
    *b=tam;
}

main()
{int a,b;
printf("Nhap a=");
scanf("%d",&a);
printf("Nhap b=");
scanf("%d",&b);
Swap1(a,b);
Swap1(a,b);
Swap2(a,b);
Swap3(a,b);
Swap4(&a,&b);
printf("Gia tri a sau khi hoan doi %d\n",a);
printf("Gia tri b sau khi hoan doi %d\n",b);
}
```

13. BÀI TẬP 13

Viết chương trình giải phương trình bậc 2 ($ax^2 + bx + c = 0$, $a \neq 0$) sử dụng con trỏ.

Chương trình

```
#include<bits/stdc++.h>
float ptb2(float a, float b, float c, float *x1, float *x2);

main(){
```



```

float a, b, c, d, x1, x2;
scanf("%f%f%f", &a, &b, &c);
ptb2(a, b, c, &x1, &x2);
}

float ptb2(float a, float b, float c, float *x1, float *x2)
{
    float d;
    if( (a!=0) )
    {
        d=b*b-4*a*c;
        if (d>0)
        {
            *x1=(-b+sqrt(d)) / (2*a);
            *x2=(-b-sqrt(d)) / (2*a);
            printf("Phuong trinh co 2 nghiem la: %f, %f", *x1, *x2);
        }
        else if (d==0)
        {
            *x1=(-b)/(2*a);
            printf("Phuong trinh co 1 nghiem la: %f", *x1);
        }
        else printf("Phuong trinh vo nghiem");
    }

    else if( (a==0) )
    {
        *x1=(-c)/b;
        printf("Phuong trinh co 1 nghiem la: %f", *x1);
    }
}

```

14. BÀI TẬP TỰ GIẢI

Số phần tử được chọn cho tới khi đạt độ chính xác cho tất cả các bài (Bài 1- đến Bài 4) < EPS=10⁻⁶

Bài 1. Viết chương trình tính sin(x) triển khai theo chuỗi số sau:



$$\sin(x) = x/1! - x^3/3! + \dots + (-1)^n \cdot x^{2n+1}/(2n+1)!$$

Số phần tử được chọn cho tới khi đạt độ chính xác: $|x^{2n+1}/(2n+1)!| < EPS=10^{-6}$ (dùng hàm fabs trong <math.h>)

Bài 2. Viết chương trình tính cos(x) triển khai theo chuỗi số sau:

$$S = \cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots$$

Bài 3. Viết chương trình tính Sh(x) triển khai theo chuỗi số sau:

$$S = \operatorname{Sh}(x) = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + \frac{x^{2n+1}}{(2n+1)!} + \dots$$

Bài 4. Viết chương trình tính ch(x) triển khai theo chuỗi số sau:

$$S = \operatorname{ch}(x) = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{x^{2n}}{(2n)!} + \dots$$



BÀI THỰC HÀNH SỐ 3

Mảng một chiều và mảng hai chiều

I. Mảng một chiều (Mảng số)

1. BÀI TẬP 1

Viết chương trình có sử dụng hàm

- Nhập mảng một chiều nguyên với n phần tử nhập từ bàn phím
- Xuất mảng A ra màn hình

Chương trình:

```
#include <stdio.h>
#include <math.h>
void Nhap(int a[], int n)
{
    int i;
    for(i=0; i<n; i++)
    {
        printf("a[%d]=", i);
        scanf("%d", &a[i]);
    }
}
void Xuat(int a[], int n)
{
    int i;
    for(i=0; i<n; i++) printf("%4d", a[i]);
}
main()
{ int a[100];
  int n;
  printf("Nhap n");
  scanf("%d", &n);
  Nhap(a, n);
  Xuat(a, n);
}
```

2. BÀI TẬP 2

Viết chương trình có sử dụng hàm

- Nhập mảng một chiều nguyên với n phần tử nhập từ bàn phím, $n \geq 10$.



- b. Xuất mảng A ra màn hình
- c. Đảo mảng A, sau đó xuất lại mảng A.

Chương trình:

```
#include <stdio.h>
#include <math.h>
void Nhap(int a[], int n)
{
    int i;
    for(i=0; i<n; i++)
    {
        printf("a[%d]=", i);
        scanf("%d", &a[i]);
    }
}
void Xuat(int a[], int n)
{
    int i;
    for(i=0; i<n; i++) printf("%4d", a[i]);
}
void Dao(int a[], int n)
{
    int i, tam;
    for (i=0; i<n/2; i++)
    {
        tam=a[i] ;
        a[i] =a[n-i-1];
        a[n-i-1]=tam;
    }
}
main()
{
    int a[100];
    int n;
    do {
        printf("Nhap n=");
        scanf("%d", &n);
    }while (n<10);
    Nhap(a, n);
    Xuat(a, n);
    Dao(a, n);
    printf("\nMang A sau khi dao\n");
    Xuat(a, n);
}
```

3. BÀI TẬP 3

Viết chương trình nhập mảng 1 chiều A có n phần tử có sử dụng hàm (với n nhập từ bàn phím). :

- a. Xuất các phần tử đã nhập ra màn hình.



b. Sắp xếp mảng theo thứ tự tăng dần

Chương trình:

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
void Nhap(float a[], int n)
{
    int i;
    for(i=0; i<n; i++)
    {
        printf("a[%d]=", i);
        scanf("%f", &a[i]);
    }
}
void Xuat(float a[], int n)
{
    int i;
    for(i=0; i<n; i++) printf("%6.2f", a[i]);
}
void Hoandoi( float *x, float *y)
{ float tam;
  tam=*x;
  *x=*y;
  *y=tam;
}
void Sapxep( float a[], int n)
{ int i,j;
  for(i=0; i<n-1; i++)
    for(j=i+1; j<n; j++ )
      if(a[i]>a[j]) Hoandoi(&a[i], &a[j]);
}
main()
{ float a[100];
  int n;
  printf("Nhap n");
  scanf("%d", &n);
  Nhap(a, n);
  Xuat(a, n);
  Sapxep(a, n);
  printf("\n Mảng sau khi sắp xếp tăng dần\n");
  Xuat(a, n);
}
```

4. BÀI TẬP 4

Viết chương trình nhập mảng 1 chiều A nguyên có n phần tử có sử dụng hàm (với n nhập từ bàn phím). :



- Xuất các phân tử đã nhập ra màn hình.
- Tìm ước số chung lớn nhất của các số trên
- Tìm bội số chung nhỏ nhất của các số trên

Chương trình:

```
#include<stdio.h>

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

void Xuat(int a[], int n)
{
    int i;
    printf("Mang vua nhap la:\n");
    for (i=0; i<n; i++) printf("%4d",a[i]);
}

int Ucln(int a,int b)
{
    if (a%b==0) return(b);
    if (b%a==0) return(a);
    if (a==b) return(a);
    else if (a>b) return(Ucln(a%b,b));
    else return(Ucln(a,b%a));
}

int Bcnn(int a,int b)
{
    return(a*b/Ucln(a,b));
}

int Uc(int a[], int n)
{
    if (n<1) return(a[0]);
    int i,d=Ucln(a[0],a[1]);
    for (i=2; i<n; i++) d=Ucln(d,a[i]);
    return(d);
}

int Bc(int a[], int n)
{
    if (n<1) return(a[0]);
    int i,d=Bcnn(a[0],a[1]);
    for (i=2; i<n; i++) d=Bcnn(d,a[i]);
}
```




```
return(d);
}
main()
{
int n;
printf("Nhap n=");
scanf("%d",&n);
int a[100];
Nhap(a,n);
Xuat(a,n);
printf("\nUCLN cua day la:%d",Uc(a,n));
printf("\nBCNN cua day la:%d",Bc(a,n));
}
```

5. BÀI TẬP 5

Viết chương trình nhập mảng 1 chiều A nguyên có n phần tử có sử dụng hàm (với n nhập từ bàn phím). :

- Xuất các phần tử đã nhập ra màn hình.
- Xóa các phần tử âm trong mảng

Chương trình:

```
#include<stdio.h>

/*-----*/
void Nhap(int *a,int n)
{int i;
for (i=0;i<n;i++)
{printf("a[%d]=",&i);scanf("%d",&a[i]);
}
}
/*-----*/
void Xuat(int *a,int n)
{int i;
for(i=0;i<n;i++)
printf("%5d",a[i]);
}
/*-----*/
void Xoaam(int *a,int *n)
{int i,j,L;
i=0;
while(i<*n)
{if(a[i]<0)
{for(j=i;j<*n-1;j++)
a[j]=a[j+1];
(*n)--;
continue;
}
i++;
}
```



```
    }  
    }  
    /*-----*/  
main()  
{  
    int a[50],n,i,j;  
    printf("Nhap n=");scanf("%d",&n);  
    Nhap(a,n);  
    Xuat(a,n);  
    printf("\n");  
    Xoaam(a,&n);  
    Xuat(a,n);  
}
```

II. Mảng một chiều (Mảng ký tự)

1. BÀI TẬP 1

Viết chương trình nhập vào một chuỗi ký tự, sau đó đếm tần suất xuất hiện của mỗi ký tự.

Chương trình:

```
#include <string.h>  
#include <stdio.h>  
#include <ctype.h>  
  
void tanso(char s[30])  
{  
    char temp[30],c[2];  
    int i=0,j=0, d;  
    c[1]='\0';  
    temp[0]='\0';  
    while(i<=strlen(s)){  
        while(isspace(s[i]))i++;  
        if(strchr(temp,s[i])==NULL)  
        {  
            c[0]=s[i];  
            strcat(temp,c);  
            i++;  
        }  
        printf("\n Trong xau co cac ky tu: %s",temp);  
        while(j<strlen(temp))  
        {  
            d=0;  
            for(i=0;i<=strlen(s);i++)  
                if(temp[j]==s[i]) d++;  
            printf("\n Ki tu %c xuất hiện %i lần",temp[j],d);  
            ++j;  
        }  
    }  
}
```



```

    }
}

/*-----*/
main()
{
char s[256],*s1;
while(1) {
printf("\nNhap vao xau ky tu:");
gets(s);
tanso(s);
printf("\n Press  q or Q to exit");
if(!kbhit())
    if (toupper(getch())=='Q') break;
}
}

```

2. BÀI TẬP 2

Viết chương trình nhập vào một chuỗi ký tự, sau đó đếm số từ của chuỗi ký tự.

Chương trình:

```

#include <string.h>
#include <stdio.h>
#include <ctype.h>
/*-----*/
int number_word(char *s)
{
int i=0, dem=0,L;
L=strlen(s);
while(i<L){
if(!isalpha(s[i]))
do {i++;} while(s[i]==' ');
else {
dem++;
do {i++;} while(s[i]!=' ');
}
}
return dem;
}
/*-----*/
main()
{
char s[256],*s1;
while(1) {
printf("\nNhap vao xau ky tu:");

```



```
gets(s);  
printf("Xau ky tu %s",s);  
printf("\n Co %d tu trong xau ky ",number_word(s));  
printf("\n Press q or Q to exit");  
if(!kbhit())  
    if (toupper(getch())=='Q') break;  
}
```

III. Các phương pháp sắp xếp

Khái niệm bài toán sắp xếp

Sắp xếp là quá trình xử lý một danh sách các phần tử (hoặc các mẫu tin) để đặt chúng theo một thứ tự thỏa mãn một tiêu chuẩn nào đó dựa trên nội dung thông tin lưu giữ tại mỗi phần tử.

Sau đây là một số phương pháp sắp xếp thông dụng sẽ được đề cập như:

1. Chọn trực tiếp - Selection sort
2. Chèn trực tiếp - Insertion sort
3. Binary Insertion sort
4. Đổi chỗ trực tiếp - Interchange sort
5. Nổi bọt - Bubble sort
6. Shaker sort
7. Shell sort
8. Heap sort
9. Quick sort
10. Merge sort
11. Radix sort

Trong học phần tin học Đại cương yêu cầu sinh viên chỉ thực hiện một vài phương pháp sắp xếp đơn giản. Sau đây trình bày có hai phương pháp thường hay áp dụng cho Tin học Đại cương, cụ thể như sau:

II.1. Phương pháp chọn trực tiếp

Giải thuật

Ý tưởng của thuật toán chọn trực tiếp mô phỏng một trong những cách sắp xếp tự nhiên nhất trong thực tế: chọn phần tử nhỏ nhất trong N phần tử ban đầu, đưa phần tử này về vị trí đứng là đầu dãy hiện hành; sau đó không quan tâm đến nó nữa, xem dãy hiện hành chỉ còn N-1 phần tử của dãy ban đầu, bắt đầu từ vị trí thứ 2; lặp lại quá trình trên cho dãy hiện hành... đến khi dãy hiện hành chỉ còn 1 phần tử. Dãy ban đầu có N phần tử, Vậy tóm tắt ý tưởng thuật toán là thực hiện N-1 lượt việc đưa phần tử nhỏ nhất trong dãy hiện hành về vị trí đứng ở đầu dãy. Các bước tiến hành như sau :



- Bước 1: $i = 1$;
- Bước 2: Tìm phần tử $a[\min]$ nhỏ nhất trong dãy hiện hành từ $a[i]$ đến $a[n]$
- Bước 3: Hoán vị $a[\min]$ và $a[i]$
- Bước 4: Nếu $i < n-1$ thì $i = i+1$; Lặp lại Bước 2 ngược lại: thì dừng. // $n-1$ phần tử đã nằm đúng vị trí.

Chương trình:

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

void Input(int a[], int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        a[i]=rand()%1000+1;
    }
}

void Swap(int *a,int *b){
    int tmp=*a;
    *a=*b;
    *b=tmp;
}

void SelectionSort(int a[],int n )
{
    int min; // chỉ số phần tử nhỏ nhất trong dãy
    int i,j;
    for ( i=0; i<n-1 ; i++)
    {
        min = i;
        for( j = i+1; j <n ; j++)
            if (a[j] < a[min])
                min = j; // ghi nhận vị trí phần tử nhỏ nhất
        Swap(&a[min], &a[i]);
    }
}

void Output(int a[], int n)
{
    int i;
    for(int i=0;i<n;i++) printf("%10d",a[i]);
}

int main(){
    int a[10000];
    int n;
    clock_t start,end;
    printf("nhap so luong phan tu :");
    scanf("%d",&n);
    Input(a,n);
    start=clock();
```



```
SelectionSort(a,n);
end=clock();
Output(a,n);
printf("\nXung bat dau: %ld",start);
printf("\nXung ket thuc: %ld",end);
}
```

II.2. Phương pháp Quick sort

II.2. 1 Sử Quick sort có sẵn trong C

Ví dụ 1: Sắp xếp dãy số tăng dần

Chương trình:

```
#include<.h>
#include<stdlib.h>
#define MAX 100000
#include<time.h>
int A[MAX];
int n;
void Input();
void Output();
int Compare (const void * a, const void * b);

main()
{
    clock_t start,end;
    printf("\n\t\tHam Quicksort cua DEV-C");
    srand(time(NULL));
    printf("\nNhap so luong phan tu: ");scanf("%d",&n);
    Input();
    start=clock();
    qsort(A, n, sizeof(int), Compare);
    end=clock();
    Output();
    printf("\nXung bat dau: %ld",start);
    printf("\nXung ket thuc: %ld",end);
}

void Input()
{ int i;
    for(int i=0;i<n;i++)
    {
        A[i]=rand()%100000+1;
    }
}

int Compare (const void * a, const void * b)
{
    return ( *(int*)a - *(int*)b );
}
```



```
void Output()
{ int i;

    printf("\nMang sau khi sap xep: \n\t");
    for(i=0;i<n;i++)
        printf("%10d",A[i]);

}
```

Ví dụ 2: Sắp xếp dãy số giảm dần

Chương trình:

```
#include<stdio.h>
#include<stdlib.h>
#define MAX 100000
#include<time.h>
int A[MAX];
int n;
void Input();
void Output();
int Compare (const void * a, const void * b);

main()
{
    clock_t start,end;
    printf("\n\t\tHam Quicksort cua DEV-C");
    srand(time(NULL));
    printf("\nNhap so luong phan tu: ");scanf("%d",&n);
    Input();
    start=clock();
    qsort(A, n, sizeof(int), Compare);
    end=clock();
    Output();
    printf("\nXung bat dau: %ld",start);
    printf("\nXung ket thuc: %ld",end);
}

void Input()
{ int i;

    for(int i=0;i<n;i++)
    {

        A[i]=rand()%10000+1;

    }

}

int Compare (const void * a, const void * b)
{

    return ( *(int*)b - *(int*)a );

}
```



```
void Output()
{ int i;

    printf("\nMang sau khi sap xep: \n\t");
    for(i=0;i<n;i++)
        printf("%10d",A[i]);

}
```

II.2. 2 Sử Quick sort tự viết

Ví dụ 1: Sắp xếp dãy số tăng dần

Chương trình:

```
#include<stdio.h>
#include<time.h>
#include<stdlib.h>
void Input(int a[], int n)
{ int i;
    for(i=0;i<n;i++)
    {
        a[i]=rand()%1000+1;
    }
}

void Swap(int *a,int *b){
    int tmp =*a;
    *a=*b;
    *b=tmp;
}

void QuickSort(int a[], int left, int right)
{
    int i = left, j = right;
    int pivot = a[(left + right) / 2]; // Lay pivot là phan tu giua
do
    {
        // Tìm vị trí i, j cần hoán vị
        while (a[i] < pivot && i < right) i++;
        while (a[j] > pivot && j > left) j--;
        if (i <= j)
        {
            Swap(&a[i], &a[j]);
            i++; j--;
        }
    } while (i <= j);

    // Khi đó pivot sẽ chốt vị trí trong mảng
    // Ta cần gọi hàm đệ quy đến 2 mảng con bên trái pivot và bên phải pivot
    if (left < j) QuickSort(a, left, j);
    if (i < right) QuickSort(a, i, right);
}
```




```

}
void Output(int a[], int n)
{ int i;
for(int i=0;i<n;i++) printf("%10d",a[i]);
}
int main(){
    int a[10000];
    int n;
    clock_t start,end;
    printf("nhap so luong phan tu :");
    scanf("%d",&n);
    Input(a,n);
    start=clock();
    QuickSort(a,0,n-1);
    end=clock();
    Output(a,n);
    printf("\nXung bat dau: %ld",start);
    printf("\nXung ket thuc: %ld",end);
}
    
```

Ví dụ 2: Sắp xếp dãy số giảm dần

Chương trình:

```

#include<stdio.h>
#include<time.h>
#include<stdlib.h>
void Input(int a[], int n)
{ int i;
    for(i=0;i<n;i++)
    {
        a[i]=rand()%1000+1;
    }
}

void Swap(int *a,int *b){
    int tmp =*a;
    *a=*b;
    *b=tmp;
}

void QuickSort(int a[], int left, int right)
{
    int i = left, j = right;
    int pivot = a[(left + right) / 2]; // Lay pivot là phan tu giua
    do
    {
        // Tìm vị trí i, j cần hoán vị
        while (a[i] > pivot && i < right) i++;
        while (a[j] < pivot && j > left) j--;
    }
    
```



```

    if (i <= j)
    {
        Swap(&a[i], &a[j]);
        i++; j--;
    }
} while (i <= j);

// Khi đó pivot sẽ chốt vị trí trong mảng
// Ta cần gọi hàm đệ quy đến 2 mảng con bên trái pivot và bên
phải pivot
    if (left < j) QuickSort(a, left, j);
    if (i < right) QuickSort(a, i, right);
}
void Output(int a[], int n)
{ int i;
  for(int i=0;i<n;i++) printf("%10d",a[i]);
}
int main(){
    int a[10000];
    int n;
    clock_t start,end;
    printf("nhập số lượng phần tử :");
    scanf("%d",&n);
    Input(a,n);
    start=clock();
    QuickSort(a,0,n-1);
    end=clock();
    Output(a,n);
    printf("\nXung bắt đầu: %ld",start);
    printf("\nXung kết thúc: %ld",end);
}

```

IV. Mảng hai chiều

1. BÀI TẬP 1

Viết chương trình có sử dụng hàm:

- Nhập một ma trận A có n dòng và m cột
- Xuất ma trận A ra màn hình

Chương trình:

```

#include <stdio.h>
#include <math.h>
void Nhap(int a[][100], int n, int m)

```



```
{int i,j;
  for(i=0; i<n; i++)
    for(j=0; j<m; j++)
    {
      printf("a[%d][%d]",i,j);
      scanf("%d",&a[i][j]);
    }
}

void Xuat(int a[][100], int n, int m)
{int i,j;
  for(i=0; i<n; i++)
  { printf("\n");
    for(j=0; j<m;j++) printf("%4d",a[i][j]);
  }
}

main()
{int a[100][100];
  int n,m;
  printf("Nhap n=");
  scanf("%d",&n);
  printf("Nhap m=");
  scanf("%d",&m);
  Nhap(a,n,m);
  Xuat(a,n,m);
}
```

2. BÀI TẬP 2

Viết chương trình có sử dụng hàm:

- Nhập một ma trận A nguyên cấp n và nhập số nguyên k
- Xuất ma trận A ra màn hình
- Xóa cột thứ k ra khỏi ma trận A.
- Xuất ma trận A sau khi xóa cột k

Chương trình:

```
#include<stdio.h>
void Nhap(int a[][50], int n)
{
  int i,j;
  for(i=0;i<n;i++)
    for(j=0;j<n;j++)
    {
      printf("A[%d][%d]=",i,j);
      scanf("%d",&a[i][j]);
    }
}
```



```

}
}

void Xuat(int a[][50],int d,int c)
{
    int i,j;
    for(i=0;i<=d-1;i++)
    {
        for(j=0;j<=c-1;j++)
        {
            printf("%d\t",a[i][j]);
        }
        printf("\n");
        printf("\n");
    }
}

void Xoacot(int a[][50],int n,int &m,int c)
{
    if(c>=0 && c<m)
    {
        for(int i=0;i<n;i++)
            for(int j=c;j<m-1;j++)
                a[i][j]=a[i][j+1];
        m--;
    }
    if(c<0) printf("\n So k không thể nhỏ hơn 0");
    if(c>n) printf("\n So k không thể lớn hơn cấp ma tran");
}

main ()
{
    int m,k,n,a[50][50];
    printf("Nhap n= ");
    scanf("%d",&n);
    m=n;
    Nhap(a,n);
    printf("\nMa tran da nhap: \n\n");
    Xuat(a,n,n);
    printf(" nhap vao so k:");
    scanf("%d",&k);
    k--;
    Xoacot(a,n,n,k);
    printf("\nMa tran sau khi xoa cot thu %d:\n",k+1);
    Xuat(a,m,n);
}

```



BÀI TẬP TỰ GIẢI

1. Viết chương trình liệt kê các giá trị chẵn trong mảng một chiều
2. Viết chương trình kiểm tra chuỗi nhập vào có đối xứng hay không?
3. Viết chương trình liệt kê các vị trí các giá trị âm trong mảng 1 chiều thực
4. Viết chương trình nhân 2 ma trận (mxn)
5. Viết chương trình tìm ma trận nghịch đảo
6. Viết chương trình tìm phần tử lớn nhất của ma trận vuông
7. Viết chương trình kiểm tra ma trận có đối xứng không ?