

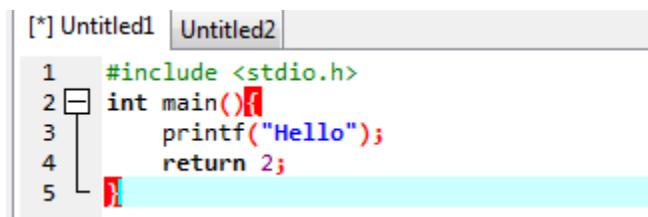
Buổi 1-25/8/2023

#1[Lập trình C]...

*Hướng dẫn sử dụng DevC++ và cách chạy chương trình...

Chương trình thường gồm 2 phần: +phần 1,khai báo thư viện

+phần 2,code



```
[*] Untitled1 [Untitled2]
1 #include <stdio.h>
2 int main(){
3     printf("Hello");
4     return 2;
5 }
```

“main()” là hàm khởi đầu,khi chạy chương trình thì các dòng code trong hàm “main()” sẽ chạy đầu tiên.

“int” là kiểu trả về,như trong hình thì khi hết thúc chương trình sẽ trả về 2(**return 2**).

#include <stdio.h> khai báo thư viện,hàm “printf” thuộc thư viện <stdio.h> không khai báo thư viện máy tính sẽ không hiểu “printf” là gì.

*kiểu dữ liệu chính trong C

Type	Size (bytes)	Format Specifier
int	at least 2, usually 4	%d , %i
char	1	%c
float	4	%f
double	8	%lf
short int	2 usually	%hd
unsigned int	at least 2, usually 4	%u
long int	at least 4, usually 8	%ld , %li
long long int	at least 8	%lld , %lli
unsigned long int	at least 4	%lu

Các kiểu chính

Số nguyên

int:4 byte

long long:8 byte

Số thực

float:4 byte

double:8 byte

Kí tự

char:1 byte

thêm “unsigned” để lấy kiểu không dấu

***Biên và quy tắc đặt tên biến**

```
#include <stdio.h>
int main(){
    kieu_du_lieu tenbien;
    printf("Hello");
    return 2;
}
```

Quy tắc đặt tên biến:-chứa chữ cái, chữ số, dấu gạch dưới

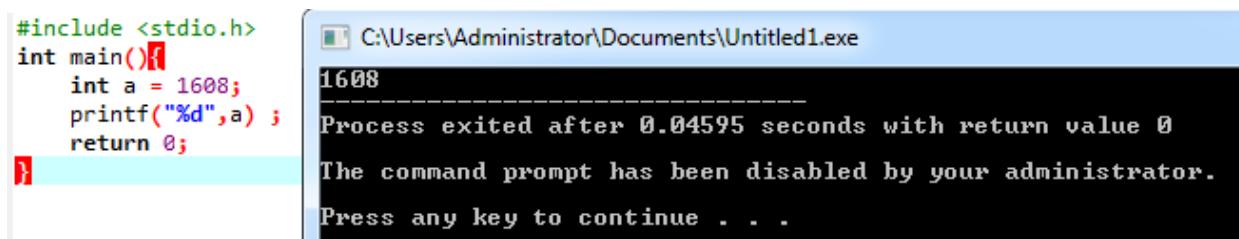
- kí tự đầu tiên không được là số
- không được trùng với các từ khoá
- không nên đặt tên biến dài
- tên biến có phân biệt chữ hoa và chữ thường
- có thể khai báo các biến cùng kiểu dữ liệu trên cùng một dòng

```
#include <stdio.h>
int main(){
    kieu_du_lieu tenbien;
    int a, b, c;
    printf("Hello");
    return 2;
}
```

*nhập và xuất trong C với printf và scanf

Type	Size (bytes)	Format Specifier
int	at least 2, usually 4	%d , %i
char	1	%c
float	4	%f
double	8	%lf
short int	2 usually	%hd
unsigned int	at least 2, usually 4	%u
long int	at least 4, usually 8	%ld , %li
long long int	at least 8	%lld , %lli
unsigned long int	at least 4	%lu

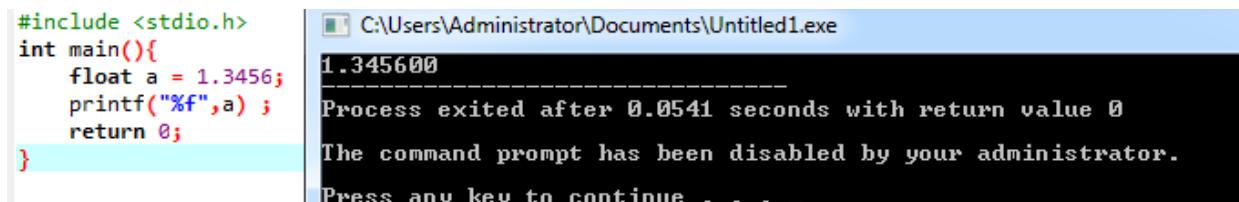
Mỗi kiểu dữ liệu đều có “Format Specifier”, để xuất được biến ra màn hình cần chọn đúng “Format Specifier”.



```
#include <stdio.h>
int main(){
    int a = 1608;
    printf("%d",a);
    return 0;
}
```

C:\Users\Administrator\Documents\Untitled1.exe
1608
Process exited after 0.04595 seconds with return value 0
The command prompt has been disabled by your administrator.
Press any key to continue . . .

Mặc định in ra sáu số đằng sau dấu chấm

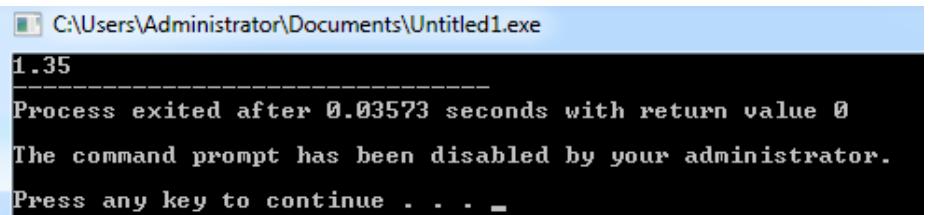


```
#include <stdio.h>
int main(){
    float a = 1.3456;
    printf("%f",a);
    return 0;
}
```

C:\Users\Administrator\Documents\Untitled1.exe
1.345600
Process exited after 0.0541 seconds with return value 0
The command prompt has been disabled by your administrator.
Press any key to continue . . .

.....

```
#include <stdio.h>
int main(){
    float a = 1.3456;
    printf("%.2f",a);
    return 0;
}
```



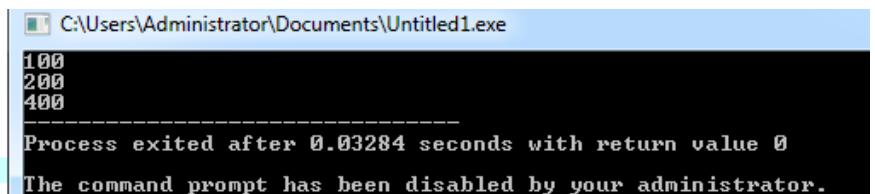
```
C:\Users\Administrator\Documents\Untitled1.exe
1.35
Process exited after 0.03573 seconds with return value 0
The command prompt has been disabled by your administrator.
Press any key to continue . . .
```

In kí tự ra màn hình

```
#include <stdio.h>
int main(){
    float a = 1.3456;
    char chucai = '0';
    printf("%c",chucai);
    return 0;
}
```

In nhiều biến cùng lúc với 1 hàm “printf”

```
#include <stdio.h>
int main(){
    float e = 1.3456;
    int a=100,b=200,c=400;
    printf("%d\n%d\n%d",a,b,c);
    return 0;
}
```

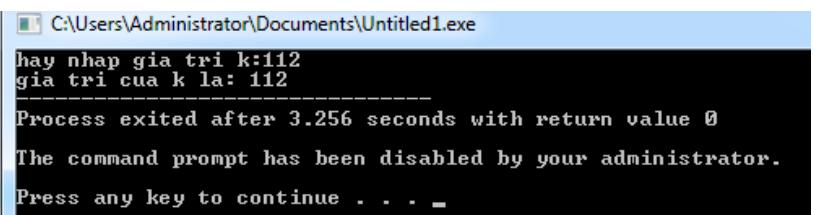


```
C:\Users\Administrator\Documents\Untitled1.exe
100
200
400
Process exited after 0.03284 seconds with return value 0
The command prompt has been disabled by your administrator.
```

Nhập giá trị từ bàn phím:

Dùng hàm “scanf”

```
#include <stdio.h>
int main(){
    int k;
    printf("hay nhap gia tri k:");
    scanf("%d", &k);
    printf("gia tri cua k la: %d", k);
    return 0;
}
```



```
C:\Users\Administrator\Documents\Untitled1.exe
hay nhap gia tri k:112
gia tri cua k la: 112
Process exited after 3.256 seconds with return value 0
The command prompt has been disabled by your administrator.
Press any key to continue . . .
```

*Toán tử trong ngôn ngữ C : Toán tử toán học, toán tử so sánh, toán tử logic, toán tử gán, toán tử tăng giảm

Toán tử trong C:+,-,* ,/,%,+,=.....

```
#include <stdio.h>
int main(){
    int a=12,b=5;
    int tong = a+b;
    int hieu = a-b;
    int tich = a*b;
    int thuong= a/b;
    int phandu= a%b;
    printf("tong: %d,hieu: %d,tich: %d,thuong: %d,phan du: %d",tong,hieu,tich,thuong);
    return 0;
}
```

Cách chia lấy số thập phân

```
#include <stdio.h>
int main(){
    int a=12,b=5;
    int tong = a+b;
    int hieu = a-b;
    int tich = a*b;
    float thuong= (float)a/b;
    int phandu= a%b;
    printf("tong: %d,hieu: %d,tich: %d,thuong: %f,phan du: %d",tong,hieu,tich,thuong);
    return 0;
}

.......
```

```
#include <stdio.h>
int main(){
    int a=100000000,b=100000;
    long long tich=(long long)a*b;
    printf("tich: %lld",tich);

    return 0;
}
```

Các toán tử “++”, “--” để tăng hay giảm 1 đơn vị

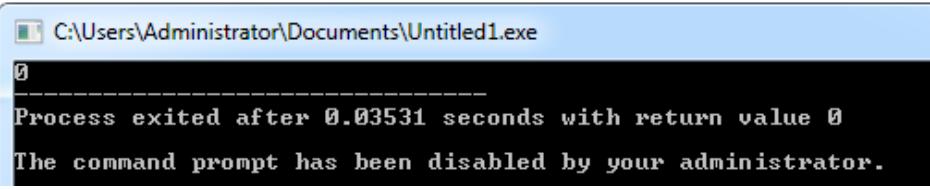
Có 2 kiểu là tăng trước và tăng sau.Với tăng trước là tăng trước rồi mới tính toán còn tăng sau là tính toán trước rồi mới tăng :))))

```
.....
//a+=b  a=a+b
//a-=b  a=a-b
//a*=b  a=a*b
//a/=b  a=a/b
//a%=b  a=a%b
```

Trong C dùng kiểu dữ liệu int thay cho kiểu dữ liệu đúng sai,với 0 là sai và các số khác là đúng

```
int main(){
    int a=100,b=101;
    int sosanh = a==b;
    printf("%d",sosanh);

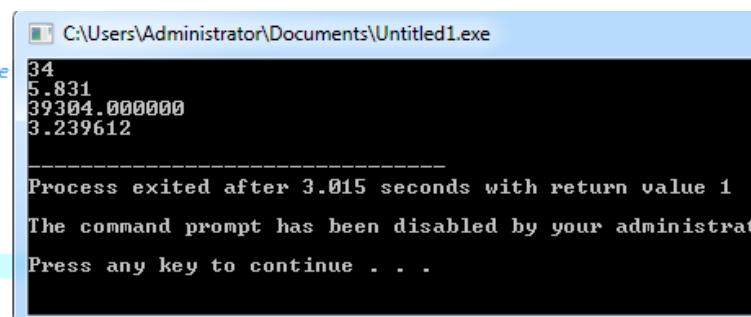
    return 0;
}
```



```
C:\Users\Administrator\Documents\Untitled1.exe
0
Process exited after 0.03531 seconds with return value 0
The command prompt has been disabled by your administrator.
```

*1 số hàm toán học trong C

```
#include <stdio.h>
#include <math.h>
//sqrt(x) tinh can bac 2 cua x tra ve double
//pow(x,y) tinh x^y tra ve double
//abs(x) tinh tri tuyet doi
//cbrt(x) tinh can bac 3 tra ve double
int main(){
    double x;
    scanf("%lf", &x);
    printf("%.3lf\n", sqrt(x));
    printf("%.3lf\n", pow(x,3));
    printf("%.3lf\n", cbrt(x));
}
```



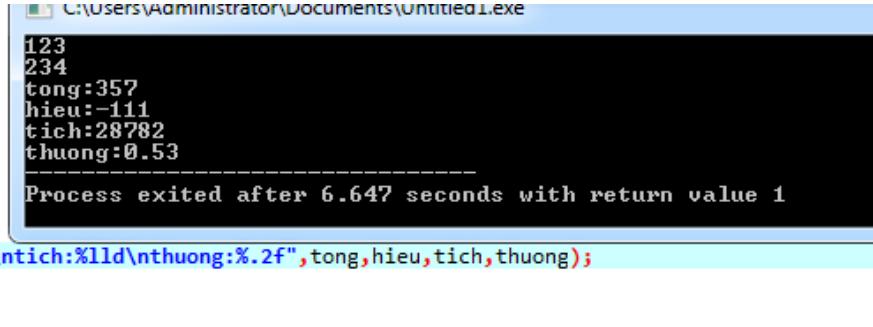
```
C:\Users\Administrator\Documents\Untitled1.exe
34
5.831
39304.000000
3.239612
Process exited after 3.015 seconds with return value 1
The command prompt has been disabled by your administrator.
Press any key to continue . . .
```

Buổi 2-26/8/2023

*Bài 1: Nhập vào 2 số nguyên, in ra tổng, hiệu, tích, thương (lấy độ chính xác với 2 chữ số).

```
#include <stdio.h>

int main(){
    int a , b;
    scanf("%d %d", &a,&b);
    int tong = a+b;
    int hieu = a-b;
    long long tich = a*b;
    float thuong= (float)a/b;
    printf("tong:%d\nhieu:%d\n\ttich:%lld\n\tthuong:%.2f",tong,hieu,tich,thuong);
    return 1;
}
```

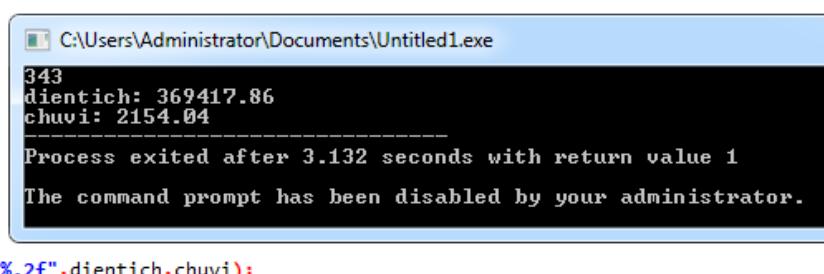


```
C:\Users\Administrator\Documents\Untitled1.exe
123
234
tong:357
hieu:-111
tich:28782
thuong:0.53
Process exited after 6.647 seconds with return value 1
```

*Bài 2: Tính chu vi, diện tích hình tròn

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

int main(){
    int r ;
    scanf("%d", &r);
    double dientich = pow(r,2)*3.14;
    float chuvi = 3.14*2*r;
    printf("dientich: %.2lf\nchuvi: %.2f",dientich,chuvi);
}
```



```
C:\Users\Administrator\Documents\Untitled1.exe
343
dientich: 369417.86
chuvi: 2154.04
Process exited after 3.132 seconds with return value 1
The command prompt has been disabled by your administrator.
```

*Bài 3: Tính khoảng cách Euclid giữa 2 điểm trong hệ tọa độ Oxy

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

int main(){
    int x1,x2,y1,y2;
    printf("nhap toa do diem 1\n");
    scanf("%d%d",&x1,&y1);
    printf("nhap toa do diem 2\n");
    scanf("%d%d",&x2,&y2);
    float khoangcach= sqrt(pow(x1-x2,2)+pow(y1-y2,2));
    printf("khoang cach giua 2 diem la: %.2f",khoangcach);

    return 1;
}
```



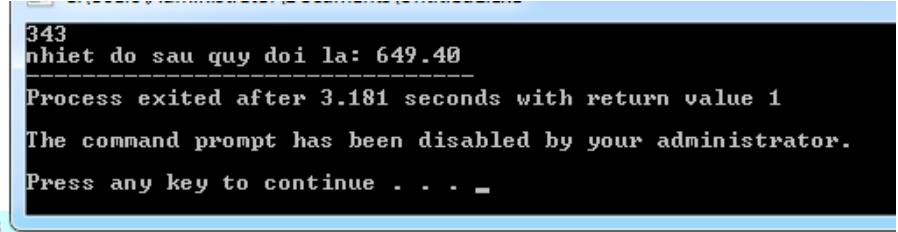
```
C:\Users\Administrator\Documents\Untitled1.exe
nhap toa do diem 1
232 34
nhap toa do diem 2
23 45
khoang cach giua 2 diem la: 209.29
Process exited after 7.978 seconds with return value 1
The command prompt has been disabled by your administrator.
```

*Bài 4: Chuyển đơn vị đo C và F

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

int main(){
    unsigned int C;
    scanf("%u",&C);
    float F=float(C)*1.8+32;
    printf("nhiet do sau quy doi la: %.2f",F);

    return 1;
}
```



```
343
nhiet do sau quy doi la: 649.40
Process exited after 3.181 seconds with return value 1
The command prompt has been disabled by your administrator.
Press any key to continue . . .
```

*Bài 5: $S_n = 1 + 2 + 3 + 4 + \dots + n$

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

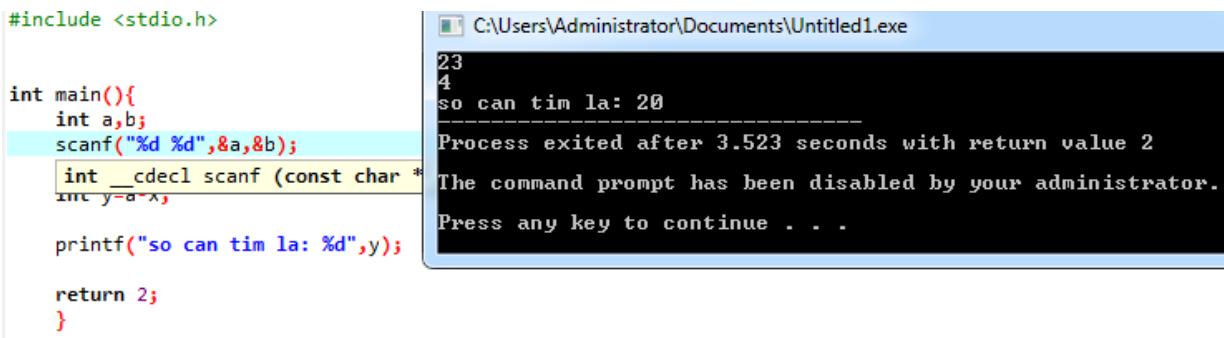
int main(){
    unsigned int n;
    scanf("%u",&n);
    int Sn= ((1+n)*n)/2;
    printf("tong cua day so la: %d",Sn);

    return 1;
}
```



```
33
tong cua day so la: 1128
Process exited after 2.212 seconds with return value 1
The command prompt has been disabled by your administrator.
Press any key to continue . . .
```

Bài 6: Cho 2 số nguyên dương a và b. Tìm số chia hết cho b lớn nhất và không vượt qua a. Chú ý không dùng vòng lặp và các hàm có sẵn.



```
#include <stdio.h>

int main(){
    int a,b;
    scanf("%d %d",&a,&b);
    int _cdecl scanf (const char *format, ...);
    int y=a-x;
    printf("so can tim la: %d",y);
    return 2;
}
```

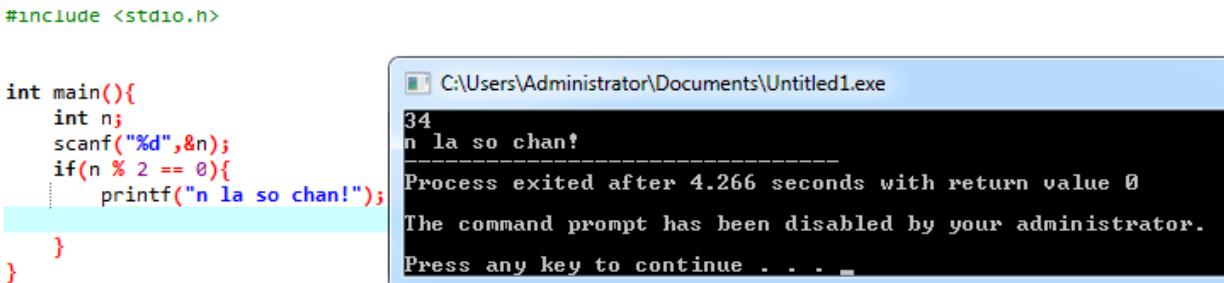
C:\Users\Administrator\Documents\Untitled1.exe
23
4
so can tim la: 20
Process exited after 3.523 seconds with return value 2
The command prompt has been disabled by your administrator.
Press any key to continue . . .

Buổi 3-27/8/2023

#2[Lập trình C]...

*Cấu trúc rẽ nhánh IF,IF ELSE

.....



```
#include <stdio.h>

int main(){
    int n;
    scanf("%d",&n);
    if(n % 2 == 0){
        printf("n la so chan!");
    }
}
```

C:\Users\Administrator\Documents\Untitled1.exe
34
n la so chan!
Process exited after 4.266 seconds with return value 0
The command prompt has been disabled by your administrator.
Press any key to continue . . .

.....

```
#include <stdio.h>

int main(){
    int n;
    scanf("%d",&n);
    if(n % 2 == 0){
        printf("\n la so chan!");
    }
    else{
        printf("\n la so le!");
    }
}
```

*Bảng mã ASCII và các thao tác kiểm tra kí tự cơ bản

A-Z:65-90 a-z:97-122 0-9:48-57

Cách kiểm tra:

```
#include <stdio.h>

int main(){
    char kitu;
    scanf("%c",&kitu);
    printf("%d",kitu);
    return 2;
}
```

Kiểm tra kí tự là chữ cái:

```
#include <stdio.h>

int main(){
    char kitu;
    scanf("%c",&kitu);
    if((kitu>=65)&&(kitu<=90)){
        printf("la chu in hoa");
    }
    else if((kitu>=97)&&(kitu<=122)){
        printf("la chu thường");
    }
}
```

Chuyển từ kí tự in hoa sang kí tự in thường và ngược lại:

```
#include <stdio.h>

int main(){
    char kitu;
    scanf("%c",&kitu);
    if(kitu>=65 && kitu<=97){
        kitu+=32;
        printf("%c",kitu);
    }
    else if(kitu>=97 && kitu<=122){
        kitu-=32;
        printf("%c",kitu);
    }
}
return 2;
}
```

Bài 16. Kiểm tra số chia hết cho 3 và 5:

```
#include <stdio.h>

int main(){
    int n ;
    scanf("%d",&n);
    if(n%3==0 && n%5==0)
        printf("thoa man yeu cau bai toan: ");
    return 2;
}
```

Bài 18. In ra số ngày của tháng

```
#include <stdio.h>

int main(){
    int t,n;
    scanf("%d%d",&t,&n);
    if(t==1 || t==3 || t==5 || t==7 || t==8 || t==10
       || t==12)
        printf("31");
    else if(t==2){
        if(n%400==0 || n%4==0 && n!=0 )
            printf("29");
        else{
            printf("28");
        }
    }
    else if(t>12){
        printf("INVALID");
    }
    else{
        printf("30");
    }
}
```

Bài 26. Số lớn, số nhỏ:

```
#include <stdio.h>

int main(){
    int a,b,c;
    scanf("%d%d%d",&a,&b,&c);
    if((a<=b && a>=c) || (a>=b && a<=c)){
        printf("%d%d",b,c);
    } else if((b<=a && b>=c) || (b>=a && b<=c)){
        printf("%d%d",a,c);
    } else if((c<=b && c>=a) || (c>=b && c<=a)){
        printf("%d%d",b,a);
    }
}
```

Bài 29. Chuyển đổi ngày sang tuần, năm, ngày

```
#include <stdio.h>

int main(){
    int ngay;
    scanf("%d",&ngay);
    int n,tuan,ngaydu1,ngaydu2;
    n = ngay/365;
    ngaydu1= ngay%365;
    tuan = ngaydu1/7;
    ngaydu2=ngaydu1%7;
    printf("nam %d tuan %d ngay %d", n,tuan,ngaydu2);
    return 2;
}
```

Bài 32. 617A

Một con voi quyết định đến thăm bạn mình. Hóa ra nhà của voi nằm ở điểm 0 và nhà của bạn anh ta nằm ở điểm x ($x > 0$) của đường tọa độ. Trong một bước, con voi có thể di chuyển 1, 2, 3, 4 hoặc 5 vị trí về phía trước. Xác định, số bước tối thiểu anh ta cần thực hiện để đến nhà của bạn mình là bao nhiêu.

Input

Dòng đầu tiên của đầu vào chứa một số nguyên x ($1 \leq x \leq 1\,000\,000$) - Tọa độ của nhà của người bạn.

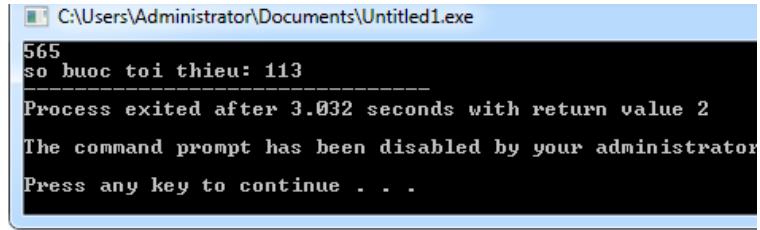
Output

In số bước tối thiểu mà voi cần thực hiện để đi từ điểm 0 đến điểm x.

```
#include <stdio.h>

int main(){
    int x ;
    scanf("%d", &x);
    int y = x/5;
    int a = y+1;
    if(x%5==0){
        printf("so buoc toi thieu: %d",y);
    }
    else{
        printf("so buoc toi thieu la: %d",a);
    }

    return 2;
}
```



Bài 33. 581A

Một ngày nọ, Vasya the Hipster quyết định đếm xem anh ta có bao nhiêu chiếc vớ.

Hóa ra anh ta có a đôi vớ đỏ và b vớ xanh.

Theo thời trang mới nhất, những người hipster sẽ mang những đôi tất có màu sắc khác nhau: một màu đỏ ở bàn chân trái, một màu xanh ở bàn chân phải.

Mỗi ngày Vasya mang vớ mới vào buổi sáng và vứt chúng đi trước khi đi ngủ vì anh không muốn giặt chúng.

Vasya tự hỏi, số ngày tối đa anh ta có thể ăn mặc theo đúng thời trang và sau đó là số ngày mà anh ta có thể đeo 2 chiếc tất cùng màu.

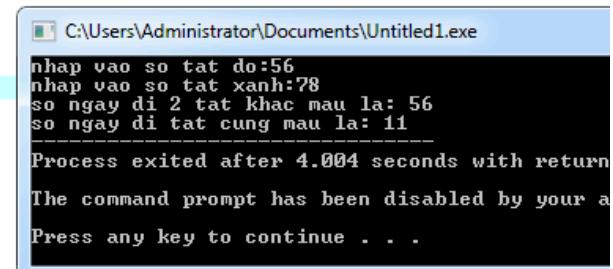
Bạn có thể giúp anh ta?

```
#include <stdio.h>

int main(){
    int r,g;

    printf("nhap vao so tat do:");
    scanf("%d",&r);
    printf("nhap vao so tat xanh:");
    scanf("%d",&g);
    if(r<g){
        printf("so ngay di 2 tat khac mau la: %d\n",r);
        printf("so ngay di tat cung mau la: %d", (g-r)/2);
    }
    else if(r>g){
        printf("so ngay di 2 tat khac mau la: %d",g);
        printf("so ngay di tat cung mau la: %d", (r-g)/2);
    }
    else{
        printf("so ngay di 2 tat khac mau la: %d",r);
        printf("so ngay di tat cung mau la: 0");
    }

    return 2;
}
```



Bài 31. Goldbach

Một cách dễ tạo ra các bài toán là học từ toán học. Bạn có thể tạo một số câu lệnh toán học ngẫu nhiên hoặc sửa đổi một số định lý để có được một cái gì đó mới và

xây dựng một nhiệm vụ mới từ đó.

Ví dụ, có một tuyên bố gọi là "phỏng đoán của Goldbach". Nó nói: "mỗi số chẵn không dưới bốn có thể được biểu diễn dưới dạng tổng của hai số nguyên tố". Hãy sửa đổi nó. Xét tuyên bố sau đây: "mỗi số nguyên không ít hơn 12 có thể được biểu diễn dưới dạng tổng của hai hợp số." Không giống như phỏng đoán của Goldbach, tôi có thể chứng minh định lý này.

Bạn được cấp một số nguyên n không nhỏ hơn 12, biểu thị nó dưới dạng tổng của hai hợp số.

Input

Dòng duy nhất chứa một số nguyên n ($12 \leq n \leq 10^6$).

Output

Xuất hai số nguyên là hợp số : x và y ($1 < x, y < n$) sao cho $x + y = n$. Nếu có nhiều giải pháp, bạn có thể xuất bất kỳ giải pháp nào.

```
#include <stdio.h>

int main(){
    int x ;
    scanf("%d", &x);

    if(x%2==0){
        printf("2 so can tim la 4 va %d",x-4);
    }
    else if(x%2!=0){
        printf("2 so can tim la 9 va %d",x-9);
    }
}
```

C:\Users\Administrator\Documents\Untitled1.exe
45
2 so can tim la 4 va 36
Process exited after 2.097 seconds with return value 2
The command prompt has been disabled by your administrator.
Press any key to continue . . .

Buổi 4-29/8/2023

#3[Lập trình C]...

*Cấu trúc rẽ nhánh Switch Case

Tương tự if-else if.

Khác elif ở,sẽ tiếp tục thực hiện các câu lệnh bên dưới nếu không có “break”.

```
#include <stdio.h>

int main(){
    int n ; scanf("%d",&n);
    switch(n){
        case 1:
            printf("31");
            break;
        case 2:
            printf("29");
            break;
        case 3:
            printf("31");
            break;
        default:
            printf("ko hop le");
    }
}
```

Có thể gộp các case lại với nhau

```
#include <stdio.h>

int main(){
    int n ; scanf("%d",&n);
    switch(n){
        case 1: case 3: case 5: case 7: case 8: case 10: case 12:
            printf("31");
            break;
        case 2:
            printf("29");
            break;
        case 4: case 6: case 9: case 11:
            printf("30");
            break;
        default:
            printf("ko hop le");
    }
}
```

Trong switch() có thể là 1 kí tự,

```
#include <stdio.h>

int main(){
    int kitu ; scanf("%c",&kitu);
    int a=12,b=2;
    switch(kitu){
        case '+':
            printf("%d",a+b);
            break;
        case '-':
            printf("%d",a-b);
            break;
        default:
            printf("ko hop le");
    }
}
```

*Câu lệnh goto trong C

Cho phép “nhảy cóc” đến các vị trí trong chương trình.

```
#include <stdio.h>

int main()
{
    printf("chao\n");
    goto vitril;
    printf("... ");
    vitril :
    printf("hello! ");
```

```
C:\Users\Administrator\Documents\Untitled1.exe

chao
hello!
-----
Process exited after 0.03153 seconds with return value 0
The command prompt has been disabled by your administrator.
Press any key to continue . . . -
```

```
#include <stdio.h>

int main(){
    vitril :
    printf("chao\n");
    goto vitril;
    printf("... ");
    printf("hello!");
```

```
C:\Users\Administrator\Documents\Untitled1.exe

chao
```

Buổi 5-30/8/2023

* Giải Thích Chi Tiết Cách Hoạt Động Vòng Lặp For Trong C

- “break”, khi vòng lặp for gặp câu lệnh “break” vòng lặp sẽ kết thúc ngay tại đó

```
#include <stdio.h>

int main()
{
    for(int i=1;i<
        printf("%d"
        if(i==3)
            break;
    }
    return 1;
}
```

```
C:\Users\Administrator\Documents\Untitled1.exe
1 2 3
-----
Process exited after 0.02727 seconds with return value 1
The command prompt has been disabled by your administrator.
Press any key to continue . . .
```

– “continue”, khi gặp câu lệnh “continue” sẽ bỏ qua vòng lặp hiện tại và tiếp tục với vòng lặp tiếp theo

```
#include <stdio.h>

int main()
{
    for(int i=1;i<
        printf("%d"
        continue;
        printf("he
    }
return 1; }
```

```
C:\Users\Administrator\Documents\Untitled1.exe
1 2 3 4
-----
Process exited after 0.03046 seconds with return value 1
The command prompt has been disabled by your administrator.

Press any key to continue . . .
```

-vòng lặp for lồng nhau.

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

```
int main()
{
    for(int i=1;i<=3;i++){
        for(int k=1;k<=5;k++)
            printf("%d %d\n",i,k);
    }
return 1; }
```

```
1 1
1 2
1 3
1 4
1 5
2 1
2 2
2 3
2 4
2 5
3 1
3 2
3 3
3 4
3 5
```

*Bài 1:

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

```
int main(){
    int n; scanf("%d", &n);
    int tong=0 ;
    for(int i=1;i<=n;i+=1){
        tong+=i;
    }
    printf("%d",tong);
    return 1;
}
```

```
C:\Users\Administrator\Documents\Untitled1.exe
6
21
Process exited after 1.963 seconds with return value 1
```

Bai 2

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

```
int main(){
    int n; scanf("%d", &n);
    int tong=0 ;
    for(int i=1;i<=n;i+=1){
        tong=tong+pow(i,2);
    }
    printf("%d",tong);
    return 1;
}
```

```
C:\Users\Administrator\Documents\Untitled1.exe
4
30
Process exited after 2.391 seconds with return value 1
The command prompt has been disabled by your administrator.
Press any key to continue . . .
```

Bai 3

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

```
int main(){
    int n; scanf("%d", &n);
    int tong=0 ;
    for(int i=0;i<=n;i+=3){
        tong=tong+i;
    }
    printf("%d",tong);
    return 1;
}
```

```
C:\Users\Administrator\Documents\Untitled1.exe
34
198
Process exited after 4.532 seconds with return value 1
The command prompt has been disabled by your administrator.
Press any key to continue . . .
```

Bai 4

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

int main(){
    int n; scanf("%d", &n);
    float tong=0 ;
    for(float i=1;i<=n;i+=1){
        tong=tong+(1/i);
    }
    printf("%.2f",tong);
    return 1;
}
```

```
9
2.83
-----
Process exited after 4.284 seconds with return value 1
The command prompt has been disabled by your administrator.
Press any key to continue . . .
```

Bai 5

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

int main(){
    int n; scanf("%d", &n);
    float tong=0 ;
    for(float i=1;i<=n;i+=1){
        tong=tong+(1/(2*i));
    }
    printf("%.2f",tong);
    return 1;
}
```

```
7
1.30
-----
Process exited after 1.757 seconds with return value 1
The command prompt has been disabled by your administrator.
Press any key to continue . . .
```

Bai 6

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

int main(){
    int n; scanf("%d", &n);
    int tong=0 ;
    for(int i=1;i<=n;i+=1){
        tong=tong+ pow(-1,i)*i;
    }
    printf("%d",tong);
}
```

```
5
-3
-----
Process exited after 3.016 seconds with return value 1
The command prompt has been disabled by your administrator.
Press any key to continue . . .
```

Bai 12

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

int main(){
    int n;
    scanf("%d", &n);
    int kqua = log10(n)+1;
    printf("so chu so la %d",kqua);

    return 1;
}
```

```
787
so chu so la 3
-----
Process exited after 3.185 seconds with return value 1
The command prompt has been disabled by your administrator.
Press any key to continue . . .
```

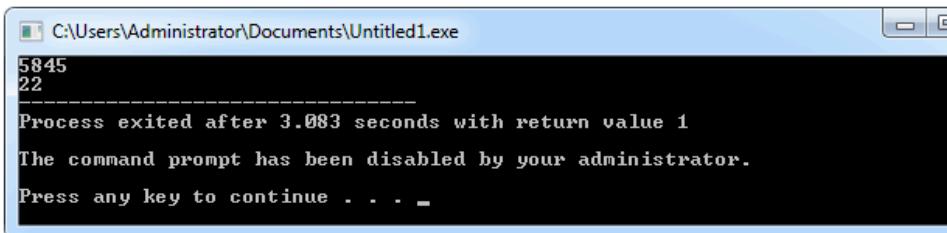
Bài 13.

Nhập vào n ($0 \leq n \leq 10^{18}$), tính tổng các chữ số của n, và in ra kết quả

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

int main(){
    int n;
    scanf("%d", &n);
    int sobitach;
    int tong = 0;
    for(;n!=0;){
        sobitach=n%10;
        tong+=sobitach;
        n=n/10;
    }
    printf("%d",tong);

    return 1;
}
```

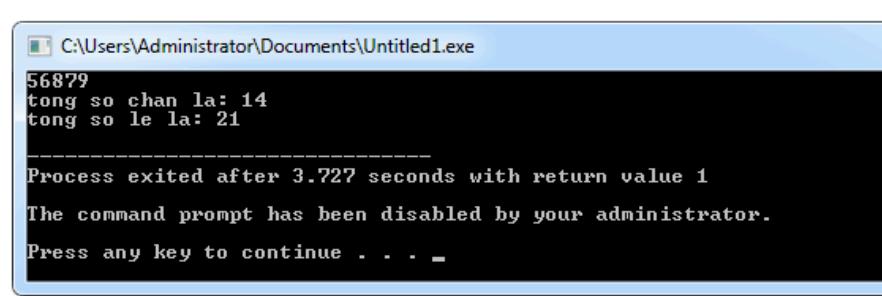


Bài 14.

Nhập vào n ($0 \leq n \leq 10^{18}$), tính tổng các chữ số là số chẵn và tổng các chữ số là số lẻ, in ra 2 tổng chẵn lẻ trên 2 dòng.

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

int main(){
    int n;
    scanf("%d", &n);
    int sobitach;
    int tong1 = 0, tong2=0;
    for(;n!=0;){
        sobitach =n%10;
        if(sobitach%2==0){
            tong2+=sobitach;
        }
        else if(sobitach%2!=0){
            tong1+=sobitach;
        }
        n=n/10;
    }
    printf("tong so chan la: %d\n",tong2);
    printf("tong so le la: %d\n",tong1);
    return 1;
}
```



Bài 15.

Nhập vào n ($0 \leq n \leq 10^{18}$), in ra số lượng chữ số chẵn, lẻ của n.

```

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

int main(){
    int n;
    scanf("%d", &n);
    int sobitach ;
    int tong1 =0 , tong2=0 ;
    for(;n!=0;){
        sobitach =n%10;
        if(sobitach%2==0){
            tong2+=1;
        }
        else if(sobitach%2!=0){
            tong1+=1;
        }
        n=n/10;
    }
    printf("tong so chan la: %d\n",tong2);
    printf("tong so le la: %d\n",tong1);
    return 1;
}

```

Bài 16.

Nhập vào n nguyên ($0 \leq n \leq 10^{18}$). Đếm số lượng chữ số của n là số nguyên tố.

```

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

int main(){
    int n;
    scanf("%d", &n);
    int sodem=0;
    int sobitach ;
    int x=0;
    for(;n!=0;x=0){ //n=2365 //n=236 //23
        sobitach=n%10;//sobitach=5 //sbt=6 // sbt=3

        for(int i=2;i<=sqrt(sobitach);i++){//sbt=5//sbt=6 sbt=3
            if(sobitach%i==0)
                x++; //x=0 //x=1
        }
        if(x==0)//x=0 //x=1
            sodem+=1; //sd=1
        n=n/10;//n=236 //n=23
    }
    printf("%d",sodem);

    return 1;
}

```

Bài 17.

Số đẹp được định nghĩa là số mà trong đó số lượng chữ số chẵn và lẻ của n

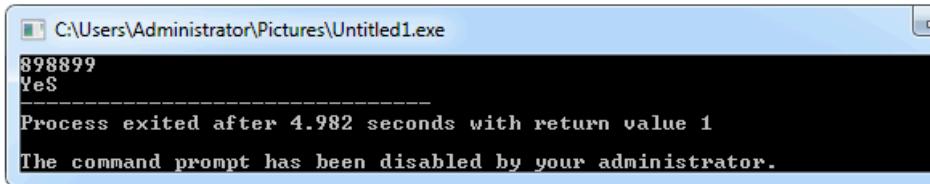
là bằng nhau. Nhập vào một số nguyên dương n ($0 \leq n \leq 10^{18}$). Nếu số nhập vào là số đẹp, thì in ra “YES”, trường hợp ngược lại in ra “NO”.

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

```
int main(){
    int chan=0, le=0;
    int sobitach;
    int n; scanf("%d", &n);
    for(;n!=0;){
        sobitach=n%10;
        if(sobitach%2==0){
            chan+=1;
        }
        else{
            le+=1;
        }
        n=n/10;
    }

    if(chan==le)
        printf("YeS");
    else
        printf("NO");

    return 1;
}
```

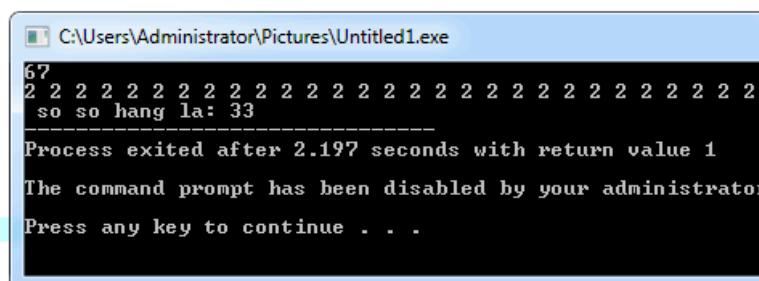


Bài 18.

Cho số nguyên dương n biểu diễn n dưới dạng tổng của các số nguyên tố sao cho số lượng số hạng trong tổng là lớn nhất.

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

```
int main(){
    int n; scanf("%d", &n);
    if(n%2==0){
        for(int i=1;i<=(n/2);i++){
            printf("%d ",2);
        }
        printf("\nso so hang la: %d",n/2);
    }
    else{
        for(int j=2;j<=n/2;j++){
            printf("%d ",2);
        }
        printf("3");
        printf("\n so so hang la: %d",n/2);
    }
}
return 1;
```



Bài 19.

Olesya thích các số gồm n chữ số và Rodion chỉ thích các số chia hết cho t.

Tìm một số số thỏa mãn cả hai.

Nhiệm vụ của bạn là: cho n và t in một số nguyên lớn hơn 0 bao gồm n chữ số chia hết cho t. Nếu số đó không tồn tại, in - 1.

Input

Dòng đơn chứa hai số, n và t ($1 \leq n \leq 100$, $2 \leq t \leq 10$) - độ dài của số và số nên chia hết cho t. Output

In một số dương như vậy mà không có số 0 đứng đầu, - câu trả lời cho vấn đề, hoặc - 1, nếu số đó không tồn tại. Nếu có nhiều câu trả lời có thể, bạn được phép in bất kỳ câu trả lời nào.

```
#include <stdio.h>
#include <math.h>
int main(){
    int n; scanf("%d", &n);
    int t; scanf("%d", &t);
    for(long long i=pow(10,n-1);i<=pow(10,n)-1;i++){
        if(i%t==0){
            printf("%lld\n",i);
        }
        else{
            continue;
        }
    }
    return 1;
}
```

```
3
9
108
117
126
135
144
153
162
171
180
189
198
207
216
225
234
243
252
261
270
279
288
297
306
315
324
333
342
351
360
369
```

Mishka là một con gấu bắc cực nhỏ. Như đã biết, những chú gấu nhỏ thích dành thời gian rảnh của mình để chơi xúc xắc cho sôcôla. Một lần vào một buổi sáng nắng tuyệt vời, đi dạo quanh những khố băng, Mishka gặp người bạn Chris của mình và họ bắt đầu chơi game.

Luật chơi rất đơn giản có n vòng đấu. Trong mỗi vòng, mỗi người chơi ném một viên xí ngầu hình khối với các số khác nhau từ 1 đến 6 được viết trên mặt của nó. Người chơi có giá trị sau khi ném xúc xắc lớn hơn, sẽ thắng vòng đấu. Trong

6

trường hợp nếu giá trị xúc xắc của người chơi bằng nhau, không ai trong số họ là người chiến thắng.

Trung bình, người chơi, người chiến thắng hầu hết các vòng, là người chiến thắng trong trò chơi. Trong trường hợp nếu hai người chơi giành được số vòng giống nhau, kết quả của trò chơi là hòa.

Mishka vẫn còn rất nhỏ và không thể đếm được thắng và thua, vì vậy cô ấy yêu cầu bạn xem trận đấu của họ và xác định kết quả của nó. Xin hãy giúp cô ấy!

Input

Dòng đầu tiên chứa số nguyên n ($1 \leq n \leq 100$) - số vòng chơi.

Các dòng n tiếp theo chứa mô tả vòng thứ i trong số chúng chứa cặp số nguyên m_i và c_i ($1 \leq m_i, c_i \leq 6$) - các giá trị trên mặt xúc xắc sau khi Mishka và Chris 'lần lượt' ném vào vòng thứ i .

```

#include <stdio.h>
#include <math.h>
int main(){
    int n; scanf("%d", &n);
    int m, c; 
    int mk=0, cr=0;
    for(int i=1;i<=n;i++){
        scanf("%d %d",&m,&c );
    }
    if(m>c){
        mk+=1;
    }
    else if(m<c){
        cr+=1;
    }
    if(mk>cr){
        printf("Mishka");
    }
    else if(cr>mk){
        printf("Chris");
    }
    else{
        printf("Friendship is magic!");
    }
}

return 1;

```

Buổi 6-5/9/2023

#5 [Lập trình C]. Vòng Lặp While Và Do While Ngôn Ngữ Lập Trình C

*Vòng lặp while

```

#include <stdio.h>

int main(){
    int i=1;
    while(i<=10){
        printf("%d",i);
    }
}

```

Sẽ lặp cho đến khi điều kiện trong ngoặc sai

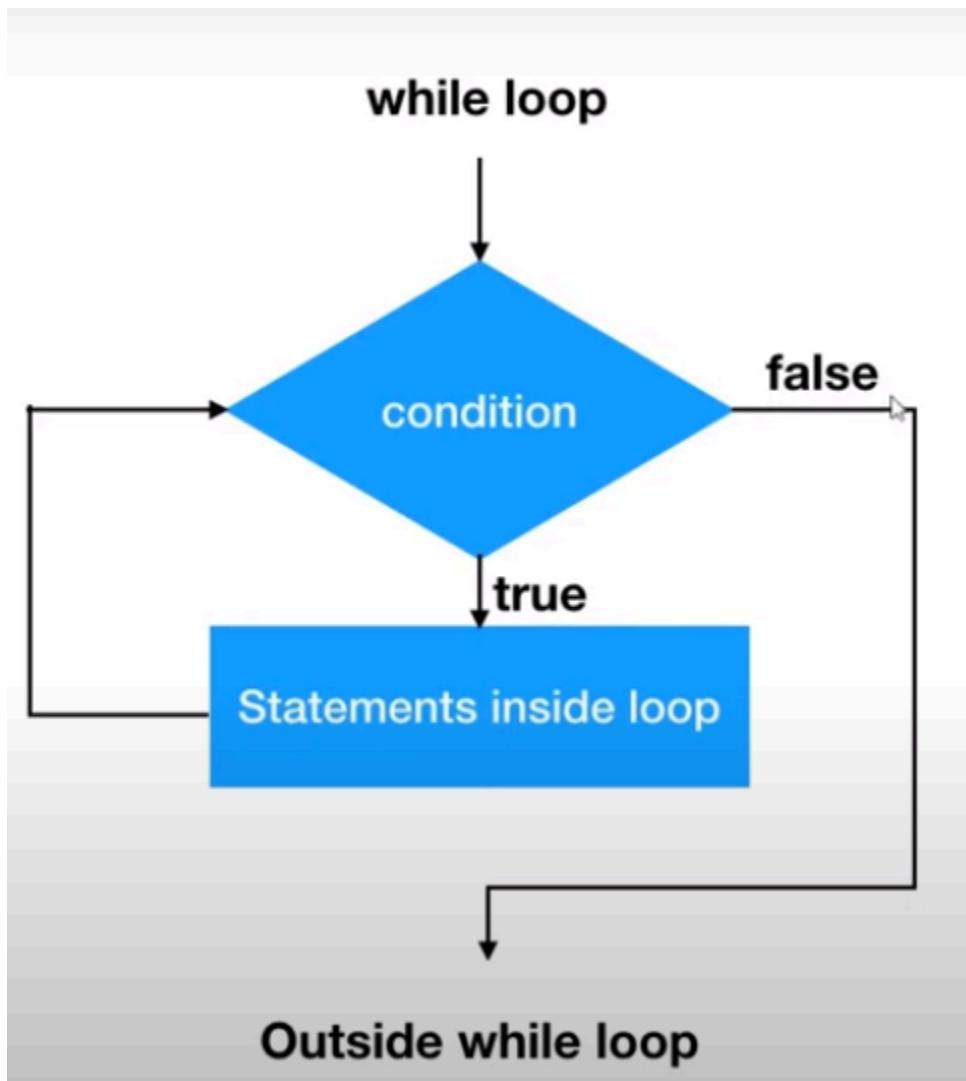
```

#include <stdio.h>

int main(){
    int i=1;
    while(i<=10){
        printf("%d ",i);
        i++;
    }
    printf("\ni sau vong lap:%d", i);
    int __cdecl printf (const char * __restrict__ _Format, ...)
}

```

Cơ chế vòng lặp while



Có thể dùng các câu lệnh như **break** và **continue**

Vòng lặp vĩnh viễn với điều kiện “1”, while(n)↔while(n!=0).

The screenshot shows a C code editor and a terminal window. The code in the editor is:

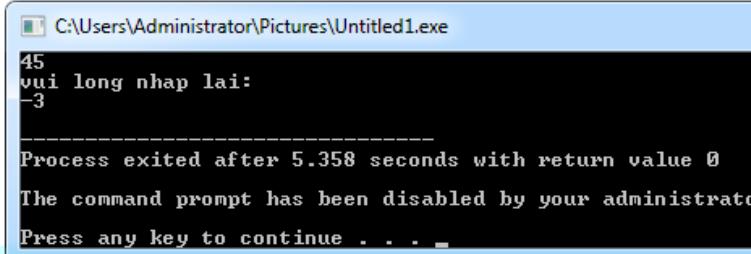
```
#include <stdio.h>
int main(){
    while(1){ // Infinite loop
        printf("vinh vien");
    }
}
```

The terminal window shows the output of the program, which is the string "vinh vien" repeated indefinitely.

*Vòng lặp do while

```
#include <stdio.h>

int main(){
    int n;
    do{
        scanf("%d", &n);
        if(n>0)
            printf("vui long nhap lai:\n");
    }while(n>0);
    return 0;
}
```



```
C:\Users\Administrator\Pictures\Untitled1.exe
45
vui long nhap lai:
-3

Process exited after 5.358 seconds with return value 0
The command prompt has been disabled by your administrator.
Press any key to continue . . .
```

Bài 25. Trailing Zero

Đếm chữ số 0 tận cùng của n giai thừa

Input

Số nguyên dương n ($1 \leq n \leq 10^9$).
).

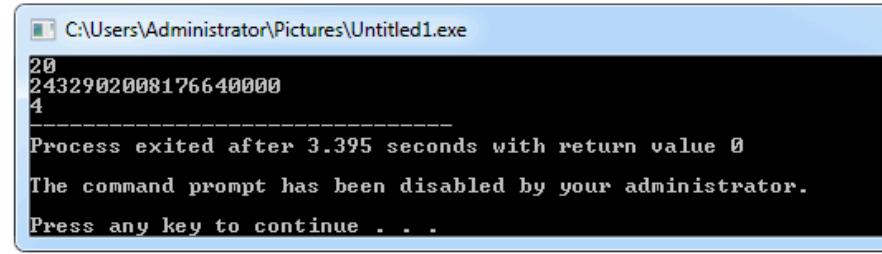
Output

Đáp án của bài toán

C1: bị hạn chế không tính được các số quá to.

```
#include <stdio.h>

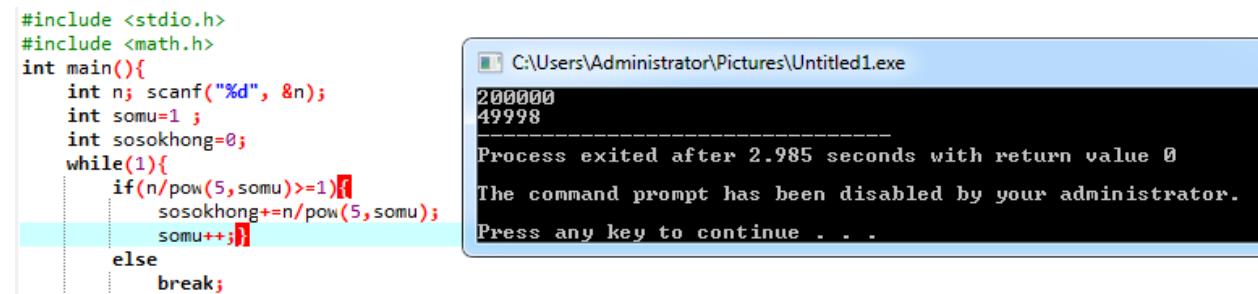
int main(){
    long long gaiithua=1;
    int n, biendem=0;
    scanf("%d", &n);
    while(n!=0){
        gaiithua*=n;
        n=n-1;
    }
    printf("%lld\n", gaiithua);
    while(1){
        if(gaiithua%10==0)
            biendem++;
        else
            break;
        gaiithua=gaiithua/10;
    }
    printf("%d", biendem);
}
```



```
C:\Users\Administrator\Pictures\Untitled1.exe
20
2432902008176640000
4

Process exited after 3.395 seconds with return value 0
The command prompt has been disabled by your administrator.
Press any key to continue . . .
```

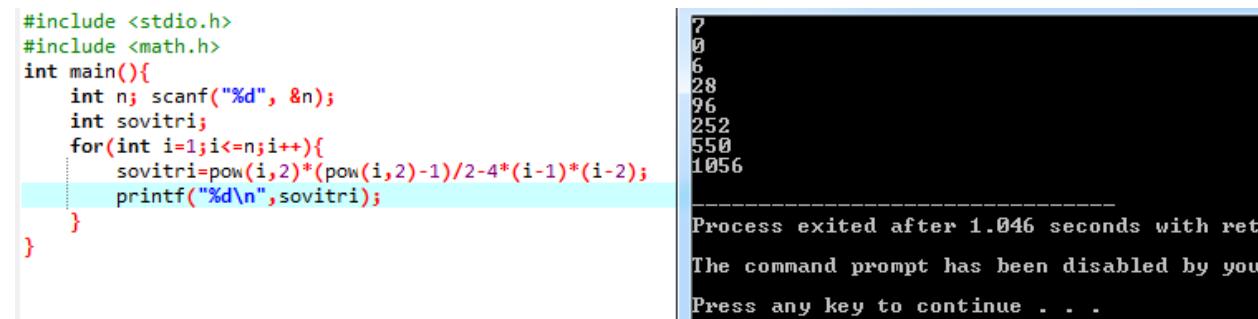
C2: tận cùng là 0 thì sẽ có tích của 1 số nguyên với 10,lại có 5^2 bằng 10⇒ tìm số lần xuất hiện của tích 5^2 ,dễ thấy số 2 xuất hiện nhiều hơn số 5 vậy bài toán quy về tìm số lần xuất hiện số 5,ở dạng 5^1 ta có số lần xuất hiện 5 là $n/5$ (lấy nguyên),ở dạng 5^2 ta có số lần xuất hiện 5^2 là $n/25$ (lấy nguyên),tương tự với 5^k ,vậy tổng số lần xuất hiện số 5= $n/5^1 + n/5^2 + \dots + n/5^K$.



```
#include <stdio.h>
#include <math.h>
int main(){
    int n; scanf("%d", &n);
    int somu=1 ;
    int sosokhong=0;
    while(1){
        if(n/pow(5,somu)>=1){
            sosokhong+=n/pow(5,somu);
            somu++;
        }
        else
            break;
    }
}
```

Bài 22. Two Knights.

Đếm số cách đặt 2 con mã trên bàn cờ vua cỡ $k \times k$ với $k = 1, 2, 3, \dots, n$ sao cho chúng không ăn nhau, 2 con mã này được coi là giống nhau.



```
#include <stdio.h>
#include <math.h>
int main(){
    int n; scanf("%d", &n);
    int sovitri;
    for(int i=1;i<=n;i++){
        sovitri=pow(i,2)*(pow(i,2)-1)/2-4*(i-1)*(i-2);
        printf("%d\n",sovitri);
    }
}
```

Ta có số cách đặt 2 quân mã trên bàn cờ là k^2C_2 ,vậy để thoả yêu cầu bài toán thì lấy tổng số vị trí có thể đặt 2 quân mã trừ đi số vị trí 2 quân mã có thể tấn công nhau,đã biết 1 quân mà thì đi hình chữ L vậy để 2 quân mã có thể tấn công nhau chúng cần nằm trong 1 ô có kích thước 3×2 ,để tạo được các ô 3×2 trên bàn cờ ta cần lấy k là số ô trên 1 hàng của bàn cờ -1 và -2 và nhân 4: $(k-1)(k-2)*2*2$.vì 3×2 tạo thành hình chữ nhật nên nhân với 2 và trong ô 3×2 có 2 vị trí mã ăn dc nhau nên nhân tiếp với 2.

#6 [Lập Trình C]. Hàm Trong Ngôn Ngữ Lập Trình C

Giúp chia nhỏ chương trình,dễ dàng hơn trong việc kiểm tra và sửa lỗi.

Ta có thể tự xây dựng 1 hàm trong ngôn ngữ C,các hàm được xây dựng sẵn như main,pow,sqrt..

-Xây dựng 1 hàm in dòng xin chao ra màn hình.

```
#include <stdio.h>

void xin_chao(){
    printf("xin chao");
}

int main(){
    xin_chao();
    void xin_chao ();
}
```

-Xây dựng hàm tính tổng các số từ 1 tới n.

```
#include <stdio.h>

void xin_chao(){
    printf("xin chao");
}
int tong(int n){
    int sum=0;
    for(int i=1;i<=n;i++) sum+=i;
    return sum;
}

int main(){
    int n=5;
    printf("%d",tong(n));
}
```

```
#include <stdio.h>

void change(int n){
    n+=10;
}

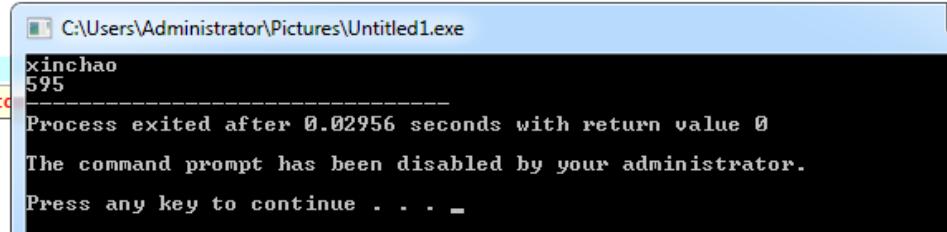
int main(){
    int x=100;
    change(x);
    printf("%d",x);
}
```

Giá trị của x không thay đổi vì khi ta ném x vào hàm change() thì giá trị của x được gán cho n rồi các câu lệnh được thực hiện với biến n.

Có thể gọi 1 hàm trong 1 hàm khác tương tự hàm main().

```
#include <stdio.h>

int tong(int n){
    int sum=0;
    for(int i=1;i<=n;i++) sum+=i;
    return sum;
}
void xin_chao(){
    printf("xinchao\n");
}
int main(){
    xin_chao();
}
```



Khi gọi 1 hàm thì cần xây dựng hàm được gọi trước khi đã khai báo nguyên mẫu hàm thì hàm xây dựng ở vị trí nào cũng được.

Nên xây dựng những hàm có thể dùng nhiều lần như hàm chinhphuong1.

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

int chinhphuong1(int n){
    int can = sqrt(n);
    if(n==can*can) return 1;
    else return 2;
}
void chinhphuong2(int n){
    int can = sqrt(n);
    if(n==can*can) printf("Yes");
    else printf("No");
}
int main(){
    |
}
```

#7[Lập trình C]. Mảng Một Chiều Và Các Bài Toán Quen Thuộc Về Mảng Trong Ngôn Ngữ Lập Trình C

.....

```
#include <stdio.h>

int main(){
    int a[12];
    scanf("%d",&a[1]);
    printf("gia tri cua phan tu thu 2 la: %d",a[1]);
    return 0;
}
```

Nhập vào các phần tử của mảng từ bàn phím.

```
#include <stdio.h>

int main(){
    int n;
    printf("nhap so luong phan tu cua mang");
    scanf("%d", &n);
    int a[n];
    for(int i=0;i<n;i++){
        printf("nhap gia tri thu %d : ", i+1 );
        scanf("%d", &a[i]);
    }
    return 0;
}
```

Có thể thấy các thay đổi đối với mảng a[] ở trong hàm nhap() vẫn được giữ nguyên

```
#include <stdio.h>

void nhap(int a[],int n){
    printf("nhap cac phan tu trong mang:\n");
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
}

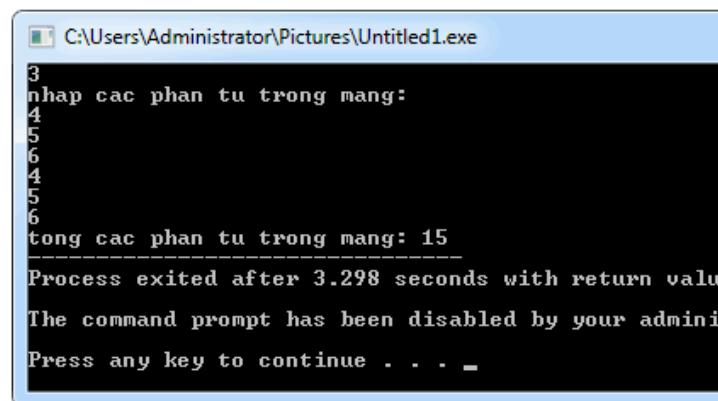
int main(){
    int n; scanf("%d",&n);
    int a[10000];
    nhap(a,n);
    for(int i=0;i<n;i++){
        printf("%d |",a[i]);
    }
    return 0;
}
```

Tính tổng các phần tử trong mảng.

```
    }
}

void in(int a[],int n){
    for(int i=0;i<n;i++){
        printf("%d\n",a[i]);
    }
}
int sum(int a[], int n){
    int tong=0;
    for(int i=0;i<n;i++){
        tong+=a[i];
    }
    return tong;
}

int main(){
    int n; scanf("%d",&n);
    int a[10000];
    nhap(a,n);
    in(a,n);
    printf("tong cac phan tu trong mang: %d",sum(a,n));
    return 0;
}
```



```
C:\Users\Administrator\Pictures\Untitled1.exe
3
nhap cac phan tu trong mang:
4
5
6
4
5
6
tong cac phan tu trong mang: 15
-----
Process exited after 3.298 seconds with return value
The command prompt has been disabled by your administrator
Press any key to continue . . .
```

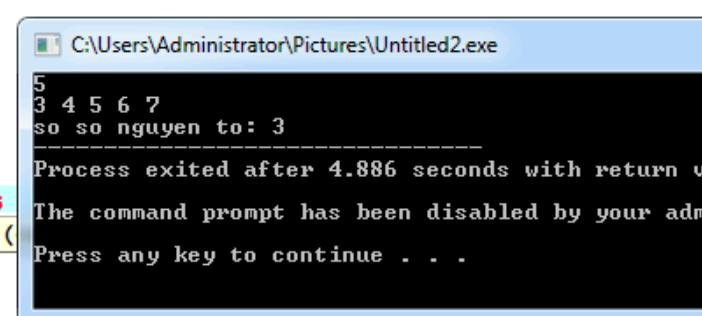
Kiểm tra số nguyên tố trong mảng.

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

int nt(int n){

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

int main(){
    int n ;scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++) scanf("%d", &a[i]);
    int dem=0;
    for(int i=0;i<n;i++){
        if(nt(a[i])) dem++;
    }
    printf("so so nguyen to: %d",dem);
}
```



```
C:\Users\Administrator\Pictures\Untitled2.exe
5
3 4 5 6 7
so so nguyen to: 3
-----
Process exited after 4.886 seconds with return value
The command prompt has been disabled by your administrator
Press any key to continue . . .
```

Kiểm tra max.

```

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

int nt(int n){

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

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

```

```

C:\Users\Administrator\Pictures\Untitled2.exe
3
1 2 4
4
Process exited after 6.798 seconds with return value 0
The command prompt has been disabled by your administrator
Press any key to continue . . .

```

Kiểm tra tính chẵn lẻ

```

#include <stdio.h>

int chan1(int n){
    if(n%2==0) return 1;
    return 0;
}

int main(){
    int chan=0;
    int le=0;
    int n; scanf("%d", &n);
    int a[n];
    for(int i=0;i<n;i++) scanf("%d", &a[i]);
    for(int i=0;i<n;i++){
        if(chan1(a[i])) chan++;
        else le++;
    }
    printf("so so chan la: %d\n",chan);
    int __cdecl printf (const char * __restrict__ __Format, ...)
}

```

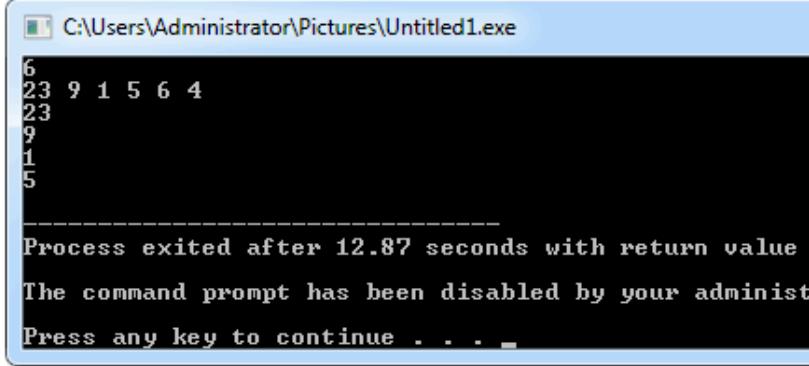
```

C:\Users\Administrator\Pictures\Untitled1.exe
4
1 3 4 6
so so chan la: 2
so so le la: 2
Process exited after 6.48 seconds with return value 0
The command prompt has been disabled by your administrator
Press any key to continue . . .

```

Liệt kê số nguyên tố

```
#include <stdio.h>
#include <math.h>
int chan1(int n){
    if(n%2==0) return 1;
    return 0;
}
int nt(int n){
    for(int i=2;i<=sqrt(n);i++){
        if(n%i==0) return 0;
        else return 1;
    }
}
```

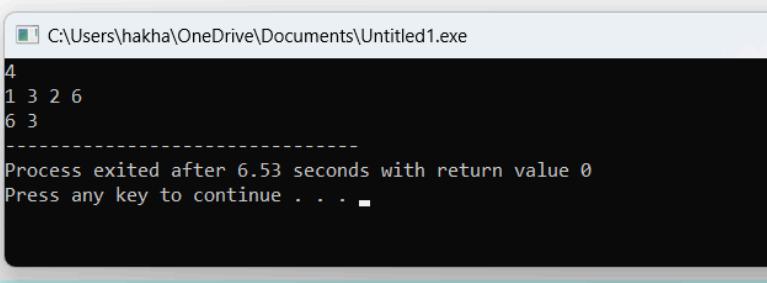


```
int main(){
    int chan=0;
    int le=0;
    int n; scanf("%d", &n);
    int a[n];
    for(int i=0;i<n;i++) scanf("%d", &a[i]);
    for(int i=0;i<n;i++){
        if(nt(a[i])) printf("%d\n",a[i]);
    }
}
```

Liệt kê số lớn thứ 2 trong mảng.

```
#include <stdio.h>

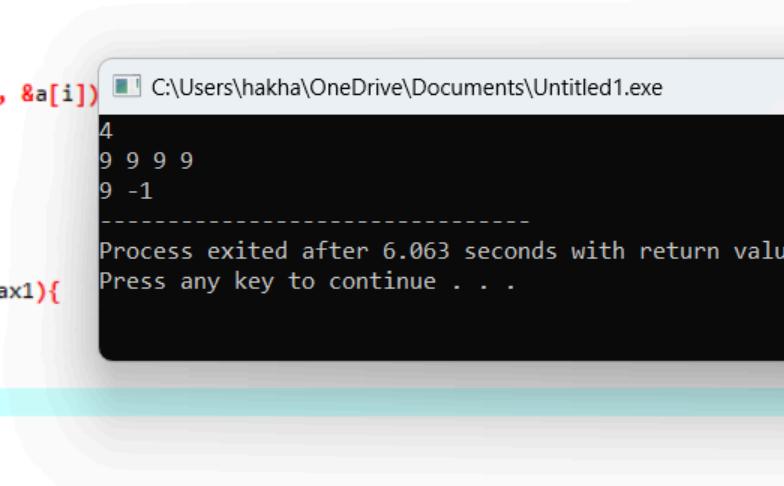
int main(){
    int n; scanf("%d", &n);
    int a[n];
    for(int i=0;i<n;i++) scanf("%d", &a[i]);
    int max1=-1e9-1,max2=-1e9-1;
    for(int i=0;i<n;i++){
        if(a[i]>max1){
            max2=max1;
            max1=a[i];
        }
        else if(a[i]>max2 && a[i]<max1){
            max2 =a[i];
        }
    }
    printf("%d %d",max1,max2);
}
```



Liệt kê số thứ 2 trong mảng,nếu ko tồn tại số lớn thứ 2 thì in ra -1.

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

```
int main(){
    int n; scanf("%d", &n);
    int a[n];
    for(int i=0;i<n;i++) scanf("%d", &a[i]);
    int max1=-1e9-1,max2=-1e9-1;
    for(int i=0;i<n;i++){
        if(a[i]>max1){
            max2=max1;
            max1=a[i];
        }
        else if(a[i]>max2 && a[i]<max1){
            max2 =a[i];
        }
    }
    if(max2 == -1e9-1) max2=-1;
    printf("%d %d",max1,max2);
```



```
C:\Users\hakha\OneDrive\Documents\Untitled1.exe
4
9 9 9 9
9 -1
-----
Process exited after 6.063 seconds with return value
Press any key to continue . . .
```

Đếm số số lẻ trong mảng.

```
#include <stdio.h>

int check(int n){
    int sobitach;
    while(n!=0){
        sobitach=n%10;
        if(sobitach%2==0) return 0;
        else return 1;
        n=n/10;
    }
}

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

In ra các số mà trong mảng luôn có 2 số lớn hơn nó

```
int check(int n){
    int sobitach;
    while(n!=0){
        sobitach=n%10;
        if(sobitach%2==0) return 0;
        else return 1;
        n=n/10;
    }
}

int main(){
    int n; scanf("%d",&n);
    int a[n]; int le=0; int b[n];
    for(int i=0;i<n;i++)
        scanf("%d", &a[i]);
    int max1=-1e9, max2=-1e9;
    for(int i=1;i<n;i++){
        if(max1<=a[i]){
            max2=max1;
            max1=a[i];
        } else if(max2<=a[i]){
            max2=a[i];
        }
    }

    for(int i;i<n;i++){
        if(a[i]<max2) printf("%d",a[i]);
    }
}
```

In ra số đẹp trong mảng nếu chứa cả chữ số 1 và 9,nếu ko chứa số đẹp in ra -1

```

] int main(){
    int n; scanf("%d",&n);
    int a[n]; int le=0; int b[n];
    for(int i=0;i<n;i++)
        |   scanf("%d", &a[i]);
    int sobitach,ded=0,c=0;
    for(int i=0;i<n;i++){
        |   if(sodep(a[i])){
            |       printf("%d ",a[i]); c++;
            |   } else if(c==0) printf("-1");
        }
    }
}

```

C:\Users\hakha\OneDrive\Documents\vscode\untitled1\main.cpp + v
5
12091 19 19 20 91
12091 19 19 91

Process exited after 13.51 seconds with return value 0
Press any key to continue . . . |

In ra số trong mảng nếu có ít nhất 1 số cạnh nó trái dấu

```

] int main(){
    int n; scanf("%d",&n);
    int a[n]; int le=0; int b[n];
    for(int i=0;i<n;i++)
        |   scanf("%d", &a[i]);
    for(int i=0;i<n;i++){
        |   if(i==0&&a[i]<0) printf("%d",a[i]);
        |   else if(i==n-1&&a[i-1]*a[n-1]<0) printf("%d",a[n-1]);
        |   else if(a[i]*a[i+1]<0||a[i]*a[i-1]<0) printf("%d",a[i]);
    }
}

```

C:\Users\hakha\OneDrive\Documents\vscode\untitled1\main.cpp + v
5
-1 2 -3 4 5
-12-34

Process exited after 12.02 seconds with return value 0
Press any key to continue . . . |

Tìm mảng đối xứng

```

#include <stdio.h>

int dx(int n,int a[]){
    int l=0,r=n-1;
    while(l<r){
        |   if(a[l]!=a[r]){
            |       return 0;
            |   l++; r--;
        }
    return 1;
}

int main(){
    int n; scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++)
        |   scanf("%d", &a[i]);
    if(dx(n,a))
        |   printf("YES");
    else
        |   printf("no");
}

```

C:\Users\hakha\OneDrive\Documents\vscode\untitled1\main.cpp + v
7
1 2 3 4 3 2 1
YES

Process exited after 14.76 seconds with return value 0
Press any key to continue . . . |

Kiểm tra mảng tăng

```
#include <stdio.h>

int main(){
    int n; scanf("%d", &n);
    int a[n];
    int cnt = 0;
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    for(int i=1;i<n;i++){
        if(a[i]>a[i-1]) cnt++;
    }
    if(cnt==n-1) printf("YES");
    else printf("NO");
}
```

```
C:\Users\hakha\OneDrive\Documents\Visual Studio Code\Projects\array> 4
1 2 3 54
YES
-----
Process exited after 5.986 seconds with return value 0
Press any key to continue . . .
```

```
#include <stdio.h>
int check(int a[],int n){
    for(int i=0;i<n-1;i++){
        if(a[i]>=a[i+1]) return 0;
    }
    return 1;
}

int main(){
    int n; scanf("%d", &n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    if(check(a,n)) printf("YES");
    else printf("NO");
}
```

```
C:\Users\hakha\OneDrive\Documents\Visual Studio Code\Projects\array> 5
1 2 3 4 6
YES
-----
Process exited after 5.446 seconds with return value 0
Press any key to continue . . .
```

Liệt kê những số không nhỏ hơn số đứng trước nó.

```
#include <stdio.h>
int check(int a[],int n){
    for(int i=0;i<n-1;i++){
        if(a[i]>=a[i+1]) return 0;
    }
    return 1;
}

int main(){
    int n; scanf("%d", &n);
    int a[n],max=-1e9;
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    for(int i=0;i<n;i++){
        if(a[i]>max){
            max=a[i];
            printf("%d ",a[i]);
        }
        else if(a[i]<max) continue;
    }
}
```

```
C:\Users\hakha\OneDrive\Documents\Visual Studio Code\Projects\array> 5
2 3 1 4 6
2 3 4 6
-----
Process exited after 4.589 seconds with return value 0
Press any key to continue . . .
```

Bài toán lineland.

```
#include <stdio.h>

int main(){
    int n; scanf("%d", &n);
    int a[n],max=-1e9;
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    for(int i=0;i<n;i++){
        if(i==0) printf("%d %d\n",a[1]-a[0],a[n-1]-a[0]);
        else if(i==n-1) printf("%d %d\n",a[n-1]-a[n-2],a[n-1]-a[0]);
        else if(i>0&&i<n-1)
            if(i>(n-1-i)) printf("%d %d\n",a[i]-a[i-1],a[i]-a[0]);
            else if(i<(n-1-i)) printf("%d %d\n",a[i]-a[i-1],a[n-1]-a[i]);
    }
}
```

```
C:\Users\hakha\OneDrive\Documents\Visual Studio Code\Projects\array> 7
1 76
1 75
2 73
1 4
71 75
1 76
-----
Process exited after 12.88 seconds with return value 0
Press any key to continue . . .
public int __cdecl printf (const char *
```

Bài toán vé xem phim.

```
#include <stdio.h>
int check(int a[],int n){
    int c25=0,c50=0;
    for(int i=0;i<n;i++){
        if(a[i]==25) c25++;
        else if(a[i]==50){
            if(c25==0) return 0;
            c25--;
            c50++;
        }
        else if(a[i]==100){
            if(c25==0 || c50*50+c25*25<75) return 0;
            if(c50!=0){
                c50--;
                c25--;
            }
            else c25-=3;
        }
    }
    return 1;
}

int main(){
    int n; scanf("%d", &n);
    int a[n],max=-1e9;
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    if(check(a,n)) printf("YES");
    else printf("NO");
}
```

```
C:\Users\hakha\OneDrive\Documents\GitHub\Algorithm\Day 1\2d.cpp:6:6: warning: variable 'n' is assigned a value of -1e+09 [-Wsign-conversion]
int a[n],max=-1e9;
          ^
Process exited after 13.31 seconds with return value 0
Press any key to continue . . . |
```

#8 [Lập Trình C]. Mảng Hai Chiều Và Các Bài Toán Áp Dụng Trong Ngôn Ngữ Lập Trình C

Cách khai báo mảng 2 chiều

```
1 #include <stdio.h>
2
3 int main(){
4     //dataType ArrayName[x][y];
5     int a[3][4];
6
7 }
```

```
#include <stdio.h>

int main(){
    //int m,n; scanf("%d%d",n,m);
    int a[3][4];
    for(int i=0;i<3;i++){
        for(int j=0;j<4;j++){
            printf("nhap cac phan tu cho hang %d cot %d",i,j);
            scanf("%d", &a[i][j]);
        }
    }
    for(int i=0;i<3;i++){
        for(int j=0;j<4;j++){
            printf("%d ",a[i][j]);
        }
    }
    printf("\n");
}
```

C:\Users\hakha\OneDrive\Do X + - □ ×

nhap cac phan tu cho hang 0 cot 9/4
nhap cac phan tu cho hang 0 cot 15
nhap cac phan tu cho hang 0 cot 26
nhap cac phan tu cho hang 0 cot 37
nhap cac phan tu cho hang 1 cot 08
nhap cac phan tu cho hang 1 cot 19
nhap cac phan tu cho hang 1 cot 20
nhap cac phan tu cho hang 1 cot 35
nhap cac phan tu cho hang 2 cot 03
nhap cac phan tu cho hang 2 cot 12
nhap cac phan tu cho hang 2 cot 23
nhap cac phan tu cho hang 2 cot 35
4 5 6 7
8 9 0 5
3 2 3 5

Process exited after 7.006 seconds with return value 0
Press any key to continue . . . |

```
#include <stdio.h>

int main(){
    int m,n; scanf("%d%d",&n,&m);
    int a[n][m];
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            printf("nhap cac phan tu cho hang %d cot %d",i,j);
            scanf("%d", &a[i][j]);
        }
    }
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            printf("%d ",a[i][j]);
        }
    }
    printf("\n");
}
```

C:\Users\hakha\OneDrive\Do X + - □ ×

4 5
nhap cac phan tu cho hang 0 cot 03
nhap cac phan tu cho hang 0 cot 14
nhap cac phan tu cho hang 0 cot 25
nhap cac phan tu cho hang 0 cot 36
nhap cac phan tu cho hang 0 cot 47
nhap cac phan tu cho hang 1 cot 08
nhap cac phan tu cho hang 1 cot 19
nhap cac phan tu cho hang 1 cot 22
nhap cac phan tu cho hang 1 cot 33
nhap cac phan tu cho hang 1 cot 44
nhap cac phan tu cho hang 2 cot 05
nhap cac phan tu cho hang 2 cot 16
nhap cac phan tu cho hang 2 cot 27
nhap cac phan tu cho hang 2 cot 38
nhap cac phan tu cho hang 2 cot 49
nhap cac phan tu cho hang 3 cot 05
nhap cac phan tu cho hang 3 cot 13
nhap cac phan tu cho hang 3 cot 22
nhap cac phan tu cho hang 3 cot 32
nhap cac phan tu cho hang 3 cot 43
3 4 5 6 7
8 9 2 3 4
5 6 7 8 9
5 3 2 2 3

Compile Log Debug Find Results Close

```
#include <stdio.h>

int main(){
    int a[2][2]={{1,2},{2,3}};
    for(int i=0 ; i<2 ; i++){
        for(int j=0 ; j<2 ; j++){
            printf("%d ",a[i][j]);
        }
    }
    printf("\n");
}
```

C:\Users\hakha\OneDrive\Do X + - □ ×

1 2
2 3

Process exited after 0.03894 seconds with return value 0
Press any key to continue . . . |

Nếu không khởi tạo thì các phần tử còn lại trong mảng sẽ bằng 0

```
#include <stdio.h>

int main(){
    int a[4][2]={{1,2},{2,3}};
    for(int i=0 ; i<4 ; i++){
        for(int j=0 ; j<2 ; j++){
            printf("%d ",a[i][j]);
        }
        printf("\n");
    }
}
```

1 2
2 3
0 0
0 0

Process exited after 0.04282 seconds with error code 0.
Press any key to continue . . . |

Tìm giá trị lớn nhất trong mảng 2 chiều.

```
int main(){
    int m,n; scanf("%d%d",&n,&m);
    int a[n][m];
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            printf("nhap cac phan tu cho hang %d cot %d",i,j);
            scanf("%d", &a[i][j]);
        }
    }
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            printf("%d ",a[i][j]);
        }
        printf("\n");
    }
    int max=-1e9-1;
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            if(a[i][j]>max) max=a[i][j];
        }
    }
    printf("%d",max);
} public int __cdecl printf (const char * __restrict__ _Format, ...)
```

nhap cac phan tu cho hang 1 cot 17
nhap cac phan tu cho hang 1 cot 29
nhap cac phan tu cho hang 1 cot 33
nhap cac phan tu cho hang 1 cot 44
nhap cac phan tu cho hang 2 cot 078
nhap cac phan tu cho hang 2 cot 132
nhap cac phan tu cho hang 2 cot 2468
nhap cac phan tu cho hang 2 cot 3367
nhap cac phan tu cho hang 2 cot 4335
nhap cac phan tu cho hang 3 cot 03332
nhap cac phan tu cho hang 3 cot 1456
nhap cac phan tu cho hang 3 cot 2245
nhap cac phan tu cho hang 3 cot 3
34
nhap cac phan tu cho hang 3 cot 4566
nhap cac phan tu cho hang 4 cot 0434
nhap cac phan tu cho hang 4 cot 1234
nhap cac phan tu cho hang 4 cot 2234
nhap cac phan tu cho hang 4 cot 324
nhap cac phan tu cho hang 4 cot 434
6 7 34 354 67
34 7 9 3 4
78 32 468 367 335
3332 456 245 34 566
434 234 234 24 34
3332

Kiểm tra và in ra số nguyên tố.

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

int main(){
    int m,n; scanf("%d%d",&n,&m);
    int a[n][m];
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            printf("nhap cac phan tu cho hang %d cot %d",i,j);
            scanf("%d", &a[i][j]);
        }
    }
    for(int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            printf("%d ",a[i][j]);
        }
    }
    printf("\n");
}
for(int i=0;i<n;i++){
    for(int j=0;j<m;j++){
        if(nt(a[i][j])) printf("%d ",a[i][j]);
    }
}
```

```
C:\Users\hakha\OneDrive\Do + v  
2 2  
nhap cac phan tu cho hang 0 cot 04  
nhap cac phan tu cho hang 0 cot 12  
nhap cac phan tu cho hang 1 cot 07  
nhap cac phan tu cho hang 1 cot 16  
4 2  
7 6  
2 7  
-----  
Process exited after 9.586 seconds with return val  
Press any key to continue . . . |
```

Kiểm tra 2 cặp số nguyên tố cùng nhau

```
#include <stdio.h>

int gcd(int a,int b){
    if(b==0)
        return a;
    return gcd(b,a%b);
}

int main(){
    int n;scanf ("%d",
    int a[n];
    for(int i=0;i<n;i++)
        scanf ("%d", &a[i]);
    long long cnt = 0;
    for(int i=0;i<n;i++)
        for(int j=i+1;j<n;j++)
            if(gcd(a[i],a[j])>1)
                cnt++;
    printf ("%lld",cnt);
}
```

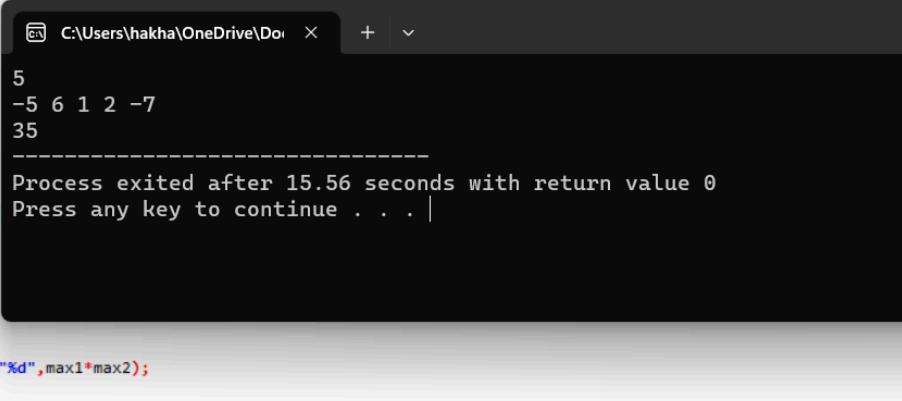
```
C:\Users\hakha\OneDrive\Do + v
5
2 3 4 5 1
9
-----
Process exited after 4.85 seconds with return value 0
Press any key to continue . . .

+){ )==1) ++cnt;
```

Tính tích 2 số lớn nhất trong mảng.

```
int main(){
    int n; scanf("%d", &n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d", &a[i]);
    }
    int max1=-1e9-1,max2=-1e9-1,min1=1e9,min2=1e9;
    for(int i=0;i<n;i++){
        if(a[i]>max1){
            max2=max1;
            max1=a[i];
        }
        else if(a[i]>max2){
            max2=a[i];
        }
    }
    for(int i=0;i<n;i++){
        if(a[i]<min1){
            min2=min1;
            min1=a[i];
        }
        else if(a[i]<min2){
            min2=a[i];
        }
    }
}

if(max1*max2>min1*min2) printf("%d",max1*max2);
else printf("%d",min1*min2);
```



5
-5 6 1 2 -7
35

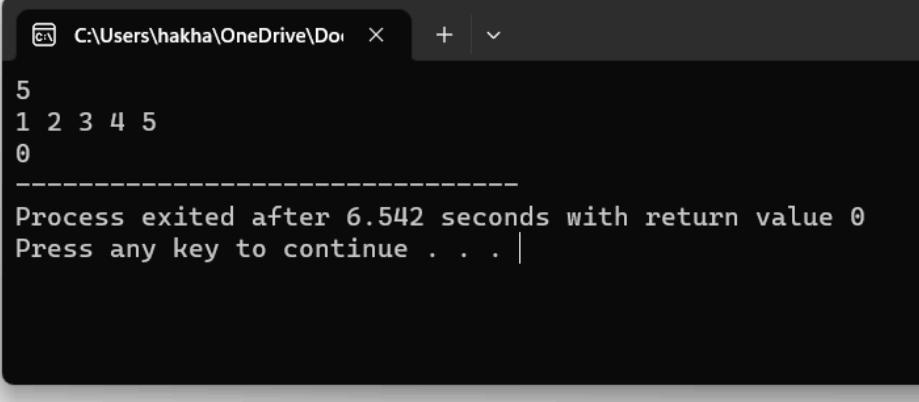
Process exited after 15.56 seconds with return value 0
Press any key to continue . . . |

Tìm dãy các số liên tiếp trái dấu nhau.

```
#include <stdio.h>

long long max(long long a,long long b){
    return a<b? b : a;
}

int main(){
    int n;scanf("%d", &n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d", &a[i]);
    }
    int res=0,cnt=1;
    for(int i=1;i<n;i++){
        if(a[i]*a[i-1]<0){
            cnt++;
        }
        else{
            cnt=1;
        }
        res=max(cnt,res);
    }
    if(res==1) printf("0");
    else printf("%d",res);
}
```



5
1 2 3 4 5
0

Process exited after 6.542 seconds with return value 0
Press any key to continue . . . |

#9 [Lập Trình C]. Con Trỏ Trong Ngôn Ngữ C | Truyền Tham Chiếu | Cấp Phát Động

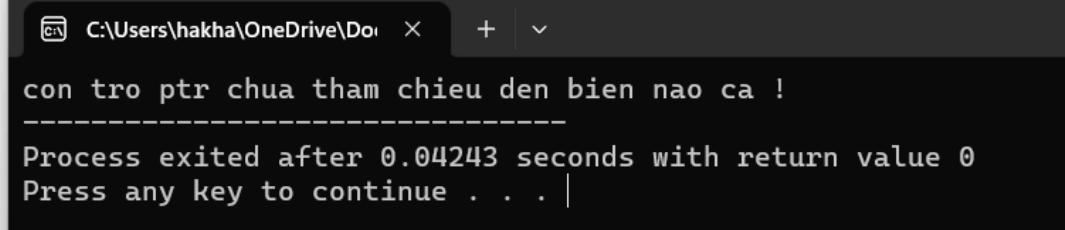
*Khái niệm, khai báo và khởi tạo con trỏ. Toán tử `địa chỉ` và toán tử `gắn tham chiếu`

Biến con trỏ không dùng để lưu 1 giá trị hằng số nào đó mà dùng để lưu `địa chỉ` của 1 ô nhớ trong bộ nhớ.

Khai báo con trỏ, nếu con trỏ chưa tham chiếu đến biến nào thì con trỏ có giá trị là `NULL`.

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

int main(){
    int a = 100;
    int *ptr;//con tro kieu int.
    if(ptr==NULL) printf("con tro ptr chua tham chieu den bien nao ca !");
}
```

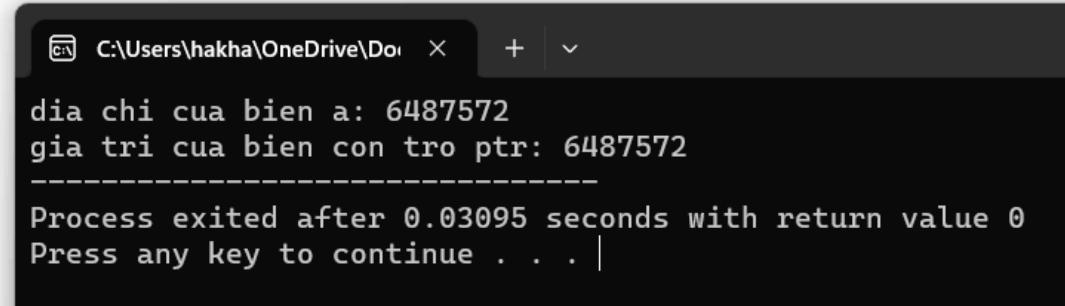


```
C:\Users\hakha\OneDrive\Documents + 
con tro ptr chua tham chieu den bien nao ca !
-----
Process exited after 0.04243 seconds with return value 0
Press any key to continue . . . |
```

Gán giá trị cho con trỏ với `địa chỉ` của biến a,"&" là toán tử có thể chỉ ra `địa chỉ` của 1 biến nào đó.lúc này có thể nói con trỏ `ptr` trả đến biến a hay con trỏ `ptr` quản lý biến a.

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

int main(){
    int a = 100;
    int *ptr;//con tro kieu int.
    ptr = &a;
    printf("dia chi cua bien a: %d\n",&a);
    printf("gia tri cua bien con tro ptr: %d",ptr);
}
```



```
C:\Users\hakha\OneDrive\Documents + 
dia chi cua bien a: 6487572
gia tri cua bien con tro ptr: 6487572
-----
Process exited after 0.03095 seconds with return value 0
Press any key to continue . . . |
```

*ptr là toán tử giải tham chiếu sẽ đưa về giá trị của biến mà con trỏ ptr đang quản lý.

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

int main(){
    int a = 100;
    int *ptr;//con trỏ kiểu int.
    ptr = &a;
    printf("dia chi cua bien a: %d\n",&a);
    printf("gia tri cua bien con trỏ ptr: %d\n",ptr);

    printf("gia tri cua bien a la: %d\n",a);
    printf("gia tri cù bien ma con trỏ ptr dang quan li %d",*ptr);

    return 0;
}

dia chi cua bien a: 6487572
gia tri cua bien con trỏ ptr: 6487572
gia tri cua bien a la: 100
gia tri cù bien ma con trỏ ptr dang quan li 100
-----
Process exited after 0.0347 seconds with return value 0
Press any key to continue . . . |
```

Khi con trỏ ptr đã tham chiếu đến biến a, thì *ptr chính là giá trị của biến a.

Sự khác nhau giữa pass by value và pass by reference. với hàm tăng 1 thì hàm này truyền giá trị của biến a vào biến bên trong hàm rồi thay đổi nên biến a không bị thay đổi, với hàm tăng 2 thì là truyền sự thay đổi tới trực tiếp địa chỉ của biến a nên biến a đã bị thay đổi sau khi hàm kết thúc.

```

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

void tang1(int a){
    a+=100;
}

void tang2(int *a){
    *a+=100;
}
int main(){
    int a=500;
    tang1(a); printf("gia tri cua bien a la %d\n",a);
    tang2(&a); printf("gia tri cua bien a la %d",a);

}

```

```

C:\Users\hakha\OneDrive\Documents\ + 
gia tri cua bien a la 500
gia tri cua bien a la 600
-----
Process exited after 0.03874 seconds with return value 0
Press any key to continue . . .

```

*a ở dưới là truy cập trực tiếp vào giá trị trên ô địa chỉ của con trỏ được đưa vào,

```

void tang2(int *a){ // ham tang2 co tham so la 1 con tro kieu int co ten la a
    *a += 100;// toan tu giao tham chieu
}

```

.....

```

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

void swap(int *a,int *b){
    int temp= *a;// gan gia tri ma cua bien ma con tro a dang cho toi cho temp
    *a=*b;//gan gia tri cua bien ma con tro a dang c tro toi cho bien ma con tro b dang tro toi
    *b=temp;//gan gia tri cho bien ma con tro b dang tro toi
}

int main(){
    int x=100,y=200;
    swap(&x,&y);
    printf("%d %d",x,y);
    public int __cdecl p
    return 0;
}

```

```

C:\Users\hakha\OneDrive\Documents\ + 
200 100
-----
Process exited after 0.03402 seconds with return value 0
Press any key to continue . . .

```

*Mối quan hệ giữa mảng 1 chiều và con trỏ trong C

Tên mảng có thể được coi như một hằng con trỏ.

Địa chỉ của mảng cũng là địa chỉ của phần tử đầu tiên.

```
#include <stdio.h>

int main(){
    int a[10]={1,2,3,4,5,6,7,8,9,0};
    printf("%d\n", a); // a la hang con tro
    printf("dia chi cua phan tu a[0] %d", &a[0]);

    return 0;
}

6487536
dia chi cua phan tu a[0] 6487536
-----
Process exited after 0.04238 seconds with return value 0
Press any key to continue . . . |
```

In ra địa chỉ của các phần tử trong mảng.

```
#include <stdio.h>

int main(){
    int a[10]={1,2,3,4,5,6,7,8,9,0};
    printf("%d\n", a); // a la hang con tro
    for(int i=0;i<10;i++){
        printf("dia chi cua phan tu a[%d] %d\n",i,&a[i]);
    }
    public int __cdecl printf (const char * __restrict__ _Format, ...)

    return 0;
}

6487536
dia chi cua phan tu a[0] 6487536
dia chi cua phan tu a[1] 6487540
dia chi cua phan tu a[2] 6487544
dia chi cua phan tu a[3] 6487548
dia chi cua phan tu a[4] 6487552
dia chi cua phan tu a[5] 6487556
dia chi cua phan tu a[6] 6487560
dia chi cua phan tu a[7] 6487564
dia chi cua phan tu a[8] 6487568
dia chi cua phan tu a[9] 6487572

-----
Process exited after 0.0417 seconds with return value 0
Press any key to continue . . . |
```

Cho con trỏ b quản lí mảng a.

```
int main(){
    int a[10]={1,2,3,4,5,6,7,8,9,0};
    int *b=a;
    for(int i=0;i<10;i++){
        printf("%d ",b[i]);
    }
}

1 2 3 4 5 6 7 8 9 0
```

Vì a là 1 hằng con trỏ nên a là con trỏ đang trỏ đến vị trí a[0],vậy a + i là con trỏ trỏ đến vị trí a[i].

```
#include <stdio.h>

int main(){
    int a[5];
    for(int i=0;i<5;i++){
        scanf("%d",a+i);
    }
    for(int i=0;i<5;i++){
        printf("%d",a[i]);
    }

    return 0;
}

1 2 3 4 5
12345
-----
Process exited after 3.323 seconds with return value 0
Press any key to continue . . . |
```

//&a[i] : (a + i)
//a[i] : *(a + i)

*Cấp phát động trong C

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

int main(){
    int *ptr = (int*)malloc(10000000 * sizeof(int));
    int n=5;
    for(int i=0;i<n;i++){
        scanf("%d",&ptr[i]);
    }
    for(int i=0;i<n;i++){
        printf("%d",*(ptr+i));
    }
    return 0;
}

4 5 6 7
3
45673
-----
Process exited after 6.527 seconds with return value 0
Press any key to continue . . . |
```

.... printf("%d",*(
})
free(ptr);
return 0;

giải phóng bộ nhớ đã xin cấp phát.

*Bài tập về mảng(tiếp).

in ra các phần tử xuất hiện trong mảng,mỗi phần tử chỉ in một lần.

```
#include <stdio.h>

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

In ra số xuất hiện nhiều nhất và tần số xuất hiện của số đó,nếu có nhiều số thì lấy số bé hơn

```
#include <stdio.h>
int cnt[1000001]={0};

int main(){
    int n; scanf("%d", &n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int res;
    int dem=0;
    for(int i=0;i<n;i++){
        cnt[a[i]]++;
    }
    for(int i=0;i<n;i++){
        if(cnt[a[i]]>dem){
            res=a[i];
            dem=cnt[a[i]];
        }
        else if(cnt[a[i]]==dem){
            if(res>a[i]) res=a[i];
        }
    }
    printf("%d %d",res,dem);
}
```

In ra số xuất hiện nhiều nhất và tần số xuất hiện của số đó,nếu có nhiều số thì lấy số xuất hiện trước.

```
#include <stdio.h>
int cnt[1000001]={0};

int main(){
    int n; scanf("%d", &n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int res;
    int dem=0;
    for(int i=0;i<n;i++){
        cnt[a[i]]++;
    }
    for(int i=0;i<n;i++){
        if(cnt[a[i]]>dem){
            res=a[i];
            dem=cnt[a[i]];
        }
        else if(cnt[a[i]]==dem){
            continue ;
        }
    }
    printf("%d %d",res,dem);
}
```



```
#include <stdio.h>

int main(){
    int n; scanf("%d", &n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int res;
    int dem=0;
    for(int i=0;i<n;i++){
        int cnt=1;
        for(int k=i+1;k<n;k++){
            if(a[i]==a[k]) cnt++;
        }
        if(cnt>dem){
            res=a[i];
            dem=cnt;
        }
    }
    printf("%d %d",res,dem);
}
```

Liệt kê các phần tử trong mảng và số lần xuất hiện.

```
#include <stdio.h>
int cnt[1000001]={0};

int main(){
    int n; scanf("%d", &n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int dem=0;
    int res;
    for(int i=0;i<n;i++){
        cnt[a[i]]++;
    }
    for(int i=0;i<n;i++){
        if(cnt[a[i]]!=0){
            printf("%d %d \n",a[i],cnt[a[i]]);
            cnt[a[i]]=0;
        }
    }
}
```

```
6
3 3 4 5 2 3
3 3
4 1
5 1
2 1
-----
Process exited after 4.781 seconds with return value 0
Press any key to continue . . . |
```

Liệt kê các phần tử chỉ xuất hiện 1 lần trong mảng.

```
#include <stdio.h>
int cnt[1000001]={0};

int main(){
    int n; scanf("%d", &n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int dem=0;
    int res;
    for(int i=0;i<n;i++){
        cnt[a[i]]++;
    }
    for(int i=0;i<n;i++){
        if(cnt[a[i]]==1){
            printf("%d\n",a[i]);
        }
    }
}
```

```
6
2 2 3 4 5 6
3
4
5
6
-----
Process exited after 4.064 seconds with return value 0
Press any key to continue . . . |
```

Mảng cộng đồng C1,

```

#include <stdio.h>
int main(){
    int n; scanf("%d\n",&n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int q;scanf("\n%d",&q);
    for(int i=1;i<=q;i++){
        int l,r;
        scanf("%d %d",&l,&r);
        long long tong=0;
        for(int k=l-1;k<r;k++){
            tong=tong+a[k];
        }
        printf("%lld",tong);
    }
}

```

5
3 4 5 6 7
2
2 3
9
2 5
22

Process exited after 16.23 seconds with return value 0
Press any key to continue . . . |

Cách 2

```

#include <stdio.h>
int main(){
    int n; scanf("%d\n",&n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    long long tongcongdon[n];
    for(int i=0;i<n;i++){
        if(i==0) tongcongdon[0] = a[0];
        else tongcongdon[i]=a[i]+tongcongdon[i-1];
    }
    int q;scanf("\n%d",&q);
    for(int k=1;k<=q;k++){
        int l,r;
        scanf("%d %d",&l,&r);
        l--;
        r--;
        if(l==0) printf("%lld\n",tongcongdon[r]);
        else printf("%lld\n",tongcongdon[r] - tongcongdon[l-1]);
    }
}

```

5
1 2 3 4 5
2
2 4
9
2 5
14

Process exited after 11.92 seconds with return value 0
Press any key to continue . . . |

Tổng dãy con lớn nhất

```

#include <stdio.h>
int main(){
    int sum1=0,sum2=0;
    int n; scanf("%d\n",&n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    for(;;){
        public int __cdecl scanf
        for(;;){
            sum1+=a[i];
            if(sum1>sum2) sum2=sum1;
            if(sum1<0) sum1=0;
        }
        printf("%d",sum2);
    }
}

```

6
1 2 -4 2 5 -2
7

Process exited after 10.03 seconds with return value 0
Press any key to continue . . . |

Tìm dãy con có k phần tử lớn nhất nếu bằng nhau thì in ra dãy sau cùng

```
#include <stdio.h>

int main(){
    int n; scanf("%d",&n);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int res;int tong2=-1e9;
    int mang[n];
    int k;scanf("%d",&k);
    for(int i=0;i<n-k+1;i++){
        mang[i]=0;
        for(int j=i;j<i+k;j++){
            mang[i]+=a[j];
        }
    }
    for(int i=0;i<n-k+1;i++){
        if(mang[i]>tong2){
            tong2=mang[i];
            res=i;
        }
        else if(mang[i]==tong2) res=i;
    }
    printf("%d\n",tong2);
    for(int i=res;i<res+k;i++){
        printf(" %d ",a[i]);
    }
}
```

Cửa sổ trượt

```
#include <stdio.h>

int main(){
    int n,k; scanf("%d%d",&n,&k);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    int res =0;
    int sum =0;
    int index =0;
    for(int i=0;i<k;i++){
        sum+=a[i];
        res=sum;
    }
    for(int i=1;i<n-k+1;i++){
        sum+=a[i+k-1]-a[i-1];
        if(sum>res){
            res=sum;
            index=i;
        }
    }
    printf("%d\n",res);
    for(int i=index;i<index+k;i++){
        printf(" %d ",a[i]);
    }
    public int __cdecl printf(
```

In ra số đầu tiên bị lặp,nếu không có in -1

```

#include <stdio.h>
#include <ctype.h>
#include <string.h>
int cnt[100001]={0};

int main(){
    int t;scanf("%d",&t);
    for(int i=1;i<=t;i++){
        int n;scanf("%d",&n);
        int a[n];
        for(int i=0;i<n;i++){
            scanf("%d",&a[i]);
        }
        int dem=0;
        for(int i=0;i<n;i++){
            if(cnt[a[i]]==1){
                printf("%d\n",a[i]);
                dem=1;
                break;
            }
            cnt[a[i]]=1;
        }
        if(dem==0) printf("-1\n");
        memset(cnt,0,sizeof(cnt));
    }
}

```

C:\Users\hakha\OneDrive\Documents\Untitled1.exe

```

2
5
2 3 4 2 3
2
3
1 2 3
-1
-----
Process exited after 19.02 seconds with return value 0
Press any key to continue . . .

```

timf giao vaf hopw

```

#include <stdio.h>

int main(){
    int n,m;scanf("%d%d",&n,&m);
    int a[n],b[m];
    for(int i=0;i<n;i++) scanf("%d",&a[i]);
    for(int i=0;i<m;i++) scanf("%d",&b[i]);
    int i=0,j=0,int giao[n],hop[n+m];int cnt1=0,cnt2=0;
    while(i<n && j<m){
        if(a[i]<=b[j]){
            hop[cnt1]=a[i];
            cnt1++;
            i++;
        }
        else if(a[i]==b[j]){
            hop[cnt1]=a[i];
            giao[cnt2]=a[i];
            i++;j++;
            cnt1++;cnt2++;
        }
        else{
            hop[cnt1]=b[j];
            cnt1++;
            j++;
        }
    }
    while(i<n){
        hop[cnt1]=a[i];
        cnt1++;
        i++;
    }
    while(j<m){
        hop[cnt1]=b[j];
        cnt1++;
        j++;
    }
    for(int k=0;k<cnt1;k++) printf("%d",hop[k]);
    printf("\n");
}

```

C:\Users\Admin\Documents\Untitled1.exe

```

4 5
2 4 5 6
1 2 4 5 7
124567
245
-----
Process exited after 11.17 seconds with return value 0
Press any key to continue . .

```

sắp xếp chẵn trước lẻ sau, theo thứ tự tăng dần

```
#include <stdio.h>

int main(){
    int n;scanf("%d",&n);
    int a[n];
    int chan[n];int le[n];
    int c=0, l=0;
    for(int i=0;i<n;i++){
        int x; scanf("%d",&x);
        if(x%2==0){
            chan[c]=x;
            c++;
        }
        else{
            le[l]=x;
            l++;
        }
    }
    for(int i=0;i<c;i++){
        int n=i;
        for(int j=i+1;j<c;j++){
            if(chan[j]<chan[n]) n=j;
        }
        int tmp=chan[n];
        chan[n]=chan[1];
        chan[1]=tmp;
    }
    for(int i=0;i<l;i++){
        int n=i;
        for(int j=i+1;j<l;j++){
            if(le[j]<le[n]) n=j;
        }
        int tmp=le[n];
        le[n]=le[1];
        le[1]=tmp;
    }
    for(int i=0;i<n;i++) printf("%d ",chan[i]);
    for(int i=0;i<l;i++) printf("%d ",le[i]);
}
```

```
C:\Users\Admin\Documents\Untitled2.exe
10
6 7 4 1 3 5 2 8 9 10
2 4 6 8 10 1 3 5 7 9
-----
Process exited after 19.14 seconds with return value 0
Press any key to continue . . .
```

in ra số xuất hiện nhiều nhất.

```
# include <stdio.h>

int main(){
    int t;scanf("%d",&t);
    while(t--){
        int n;scanf("%d",&n);
        int a[n];
        for(int i=0;i<n;i++) scanf("%d",&a[i]);
        int cnt[30001]={0};
        for(int i=0;i<n;i++){
            cnt[a[i]]++;
        }
        int res=0;
        for(int i=0;i<n;i++){
            if(res<cnt[a[i]]) res=cnt[a[i]];
        }
        for(int i=0;i<n;i++){
            if(res==cnt[a[i]]){
                printf("%d",a[i]);
                cnt[a[i]]=0;
            }
        }
    }
}
```

```
C:\Users\Admin\Documents\Untitled1.exe
2
10
2 4 1 3 5 7 3 2 4 3
3
10
1 2 3 4 5 6 7 8 9 0
1234567890
-----
Process exited after 27.36 seconds with return value 0
Press any key to continue . . .
```

trộn 2 dãy và sắp xếp,dãy a tăng dần ở chỉ số lẻ,dãy b giảm dần ở chỉ số chẵn.

The screenshot shows a C program being run in a terminal window. The code reads two integers n and t from input. It then initializes arrays a and b with size n. It performs two passes: one where it swaps elements if a[i] < a[i+1] or b[i] > b[i+1], and another where it swaps elements if a[i] > a[i+1] or b[i] < b[i+1]. Finally, it prints the arrays. The output shows the initial values 2 and 5, followed by the sorted arrays [2, 3, 1, 4, 5] and [1, 2, 4, 5, 2].

```
#include <stdio.h>
int main(){
    int t;scanf("%d",&t);
    while(t--){
        int n;scanf("%d",&n);
        int a[n],b[n];
        for(int i=0;i<n;i++) scanf("%d",&a[i]);
        for(int i=0;i<n;i++) scanf("%d",&b[i]);
        for(int i=0;i<n;i++){
            int n=i;
            for(int j=i+1;j<n;j++){
                if(a[j]<a[n]) n=j;
            }
            int tmp=a[n];
            a[n]=a[i];
            a[i]=tmp;
        }
        for(int i=0;i<n;i++){
            int n=i;
            for(int j=i+1;j<n;j++){
                if(b[j]>b[n]) n=j;
            }
            int tmp=b[n];
            b[n]=b[i];
            b[i]=tmp;
        }
        int c=0,l=0;
        for(int i=0;i<n+n;i++){
            if(i%2==0){
                printf("%d ",a[c]);
                c++;
            }
            else{
                printf("%d ",b[l]);
                l++;
            }
        }
    }
}
```

đổi lấy số tiền ít nhất.

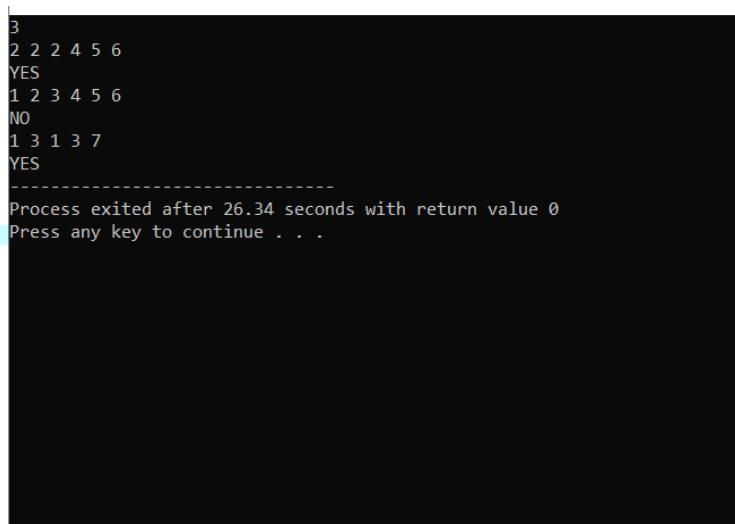
The screenshot shows a C program being run in a terminal window. The code reads an integer t and then a series of 10 integers a[10]. It then iterates through these integers, adding them to a variable cnt and taking the remainder of n/a[i]. Finally, it prints the value of cnt. The output shows the initial values 2, 7902, 11, 250, and 2.

```
# include <stdio.h>
int main(){
    int t;scanf("%d",&t);
    while(t--){
        int n;scanf("%d",&n);
        int a[10]={1000,500,200,100,50,20,10,5,2,1};
        int cnt=0;
        for(int i=0;i<10;i++){
            cnt+=n/a[i];
            n=n%a[i];
        }
        printf("%d",cnt);
    }
}
```

Dãy ưu thế.

```
#include <stdio.h>

int main(){
    int t;scanf("%d",&t);
    while(t--){
        int so=0;
        int soc=0,sol=0;
        int a[200];
        char kitu=' ';
        while(kitu!='\n'){
            int x;scanf("%d",&x);
            if(x%2==0){
                soc++;
                so++;
            }
            else{
                sol++;
                so++;
            }
            kitu=getchar();
        }
        if(so%2==0 && soc>sol) printf("YES");
        else if(so%2!=0 && sol>soc) printf("YES");
        else printf("NO");
    }
}
```



```
3
2 2 2 4 5 6
YES
1 2 3 4 5 6
NO
1 3 1 3 7
YES
-----
Process exited after 26.34 seconds with return value 0
Press any key to continue . . .
```

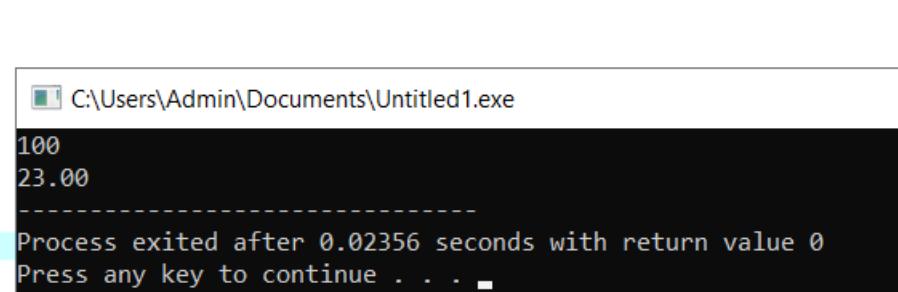
*Define và Typedef

*Typedef là một từ khoá được sử dụng để tạo nên một tên mới cho những kiểu dữ liệu của ngôn ngữ lập trình C như : int, long long, float,... hoặc những kiểu dữ liệu mà người dùng định nghĩa(user define) như struct, class.

giúp tăng tốc độ code bằng cách định nghĩa lại các kiểu dữ liệu với các tên thay thế ngắn gọn hơn.

```
#include <stdio.h>

typedef long long ll;
typedef float sothuc;
int main(){
    ll x=100;
    sothuc y=23;
    printf("%lld\n",x);
    printf("%.2f",y);
}
```



```
C:\Users\Admin\Documents\Untitled1.exe
100
23.00
-----
Process exited after 0.02356 seconds with return value 0
Press any key to continue . . .
```

*Define

Tương tự như typedef thì define cũng có thể sử dụng để định nghĩa tên mới cho kiểu dữ liệu, ngoài ra nó còn định nghĩa được cả những giá trị hoặc cú pháp.

bắt đầu bằng dấu # và kết thúc không có ;

```
#include <stdio.h>
#define ll long long
#define songuyen int
#define pi 3.142
#define FOR(i , a, b) for(int i=(a);i<(b);i++)
#define FORd(i , a, b) for(int i=(a);i>=b;i--)
} int main(){
    ll x;
    songuyen y;
    printf("%.2f\n",pi);
    FOR(i,0,5){
        printf("%d",i);
    }
    return 0;
}
```

*Hệ cơ số đếm cơ bản .

1.hệ nhị phân

với mỗi kbit,bạn có thể biểu diễn 2^k giá trị khác nhau.

chuyển từ thập phân sang nhị phân:chia cho 2,lấy phần dư và viết ngược từ dưới lên.

chuyển từ nhị phân sang thập phân lấy lần lượt các số 0,1 từ sau về trước nhân với $2^n(n=0,1,2,...)$

*chương trình chuyển từ thập phân sang nhị phân.

```
#include <stdio.h>

int main(){
    int n;
    scanf("%d",&n);
    int a[100];
    int dem=0;
    while(n){
        a[dem]=n%2;
        n=n/2;
        dem++;
    }
    for(int i=dem-1;i>=0;i--){
        printf("%d",a[i]);
    }
}
```

*Hệ 16

 28TECH
Become A Better Developer

2. Hệ 16:

 Hệ 16 được **sử dụng phổ biến** thứ 2 sau hệ nhị phân, hệ 16 sử dụng **16 kí tự** để biểu diễn số gồm : **10 chữ số từ 0 tới 9 và 6 kí tự từ A tới F**. Các số 16 thường được thêm chữ 0x ở đầu để thể hiện đây là số ở hệ 16

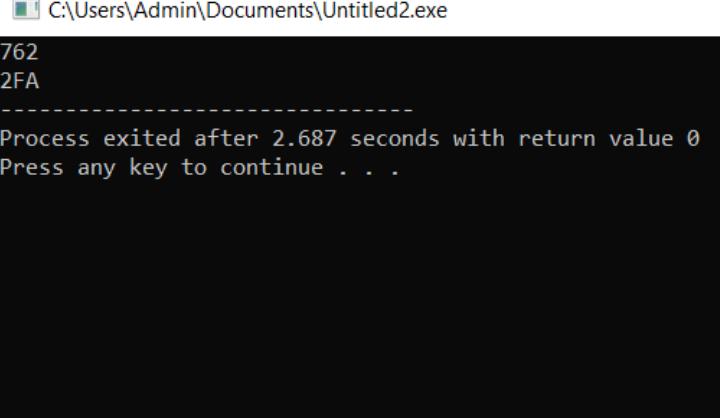
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15



chương trình chuyển từ thập phân sang hệ 16.

```
#include <stdio.h>

int main(){
    int n;
    scanf("%d",&n);
    int a[100];
    int dem=0;
    while(n){
        a[dem]=n%16;
        n=n/16;
        dem++;
    }
    for(int i=dem-1;i>=0;i--){
        if(a[i]<10 )printf("%d",a[i]);
        else printf("%c",a[i]+55);
    }
}
```


C:\Users\Admin\Documents\Untitled2.exe
762
2FA

Process exited after 2.687 seconds with return value 0
Press any key to continue . . .

*Độ phức tạp của thuật toán.

Sử dụng công thức toán học để tìm ra ngay lời giải	O(1)
Tìm kiếm nhị phân	O(logn)
Các thao tác của set, map, hàng đợi ưu tiên	O(logn)
Kiểm tra số nguyên tố, phân tích thừa số nguyên tố	O(sqrt(n))
Đọc n số từ input	O(n)
Duyệt qua mảng	O(n)
Hàm sort trong thư viện	O(nlogn)
Sàng số nguyên tố	O(nloglogn)
Duyệt các tập con cỡ k của tập có n phần tử	O(n^k)
Duyệt mọi tập con	O(2^n)
Duyệt mọi hoán vị	O(n!)

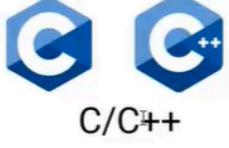
độ phức tạp về thời gian và độ phức tạp về không gian là 2 yếu tố quyết định một thuật toán có thích hợp để giải quyết vấn đề nào đó hay không

độ phức tạp về thời gian được quan tâm nhiều hơn

28TECH
Become A Better Developer

Trên các trang chấm bài online

Giới hạn thời gian của bài toán là **1 - 2s** đối với:



C/C++

Giới hạn thời gian của một bài toán là **4s** đối với:



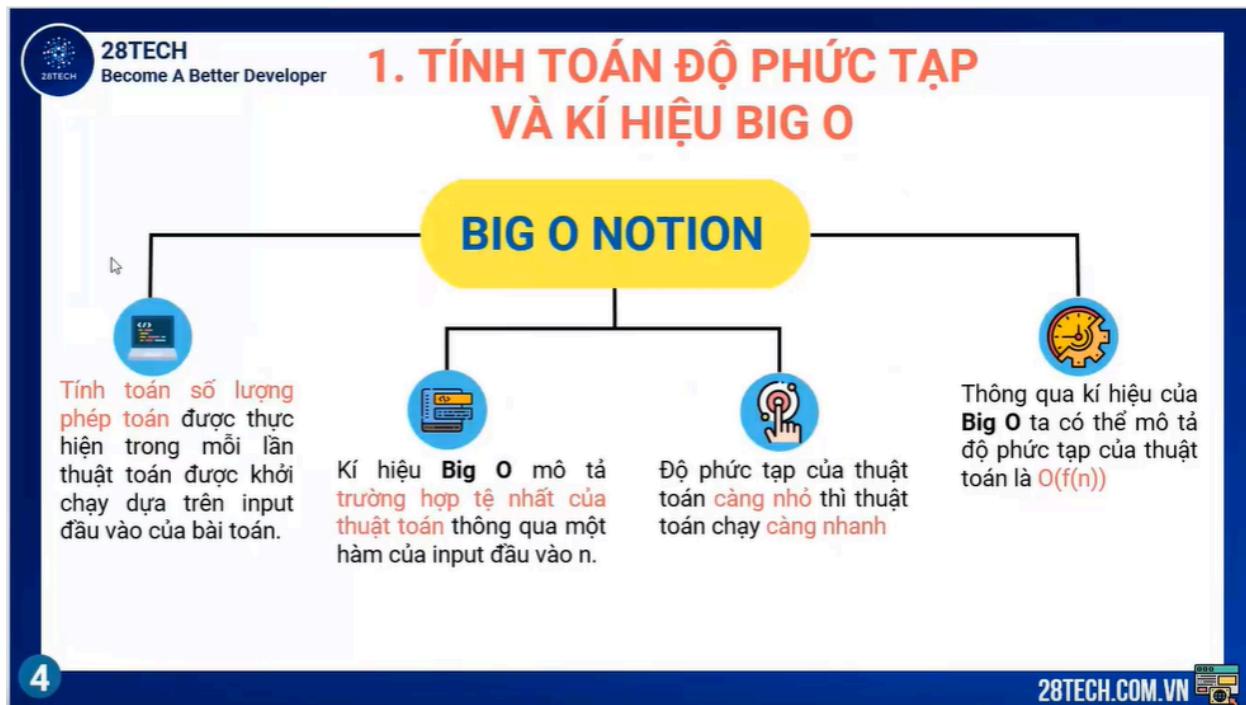
Java Python

Thông thường **1s** các bạn có thể thực hiện được từ **$10^8 - 5 \cdot 10^8$** phép toán.



3 28TECH.COM.VN

chọn bậc cao nhất của hàm để đại diện cho độ phức tạp của thuật toán.



các phép toán +,-,* ,/.. cơ bản hoặc gán,nhập,xuất,..có độ phức tạp là $O(1)$.

các căn bậc n quy chung về logn.

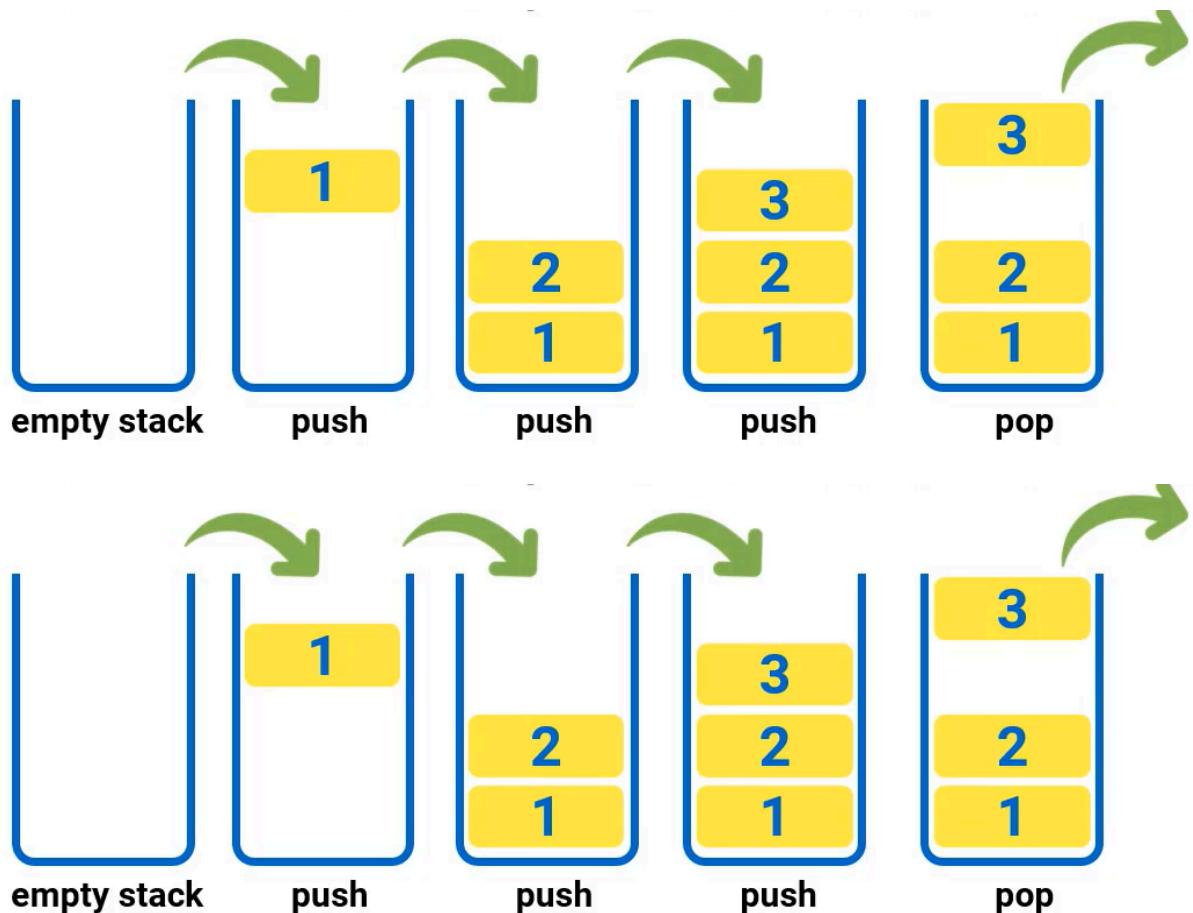
trong trường hợp thuật toán của bạn có nhiều khối thực thi thì độ phức tạp của thuật toán sẽ được xét với khối lệnh phức tạp nhất.

*Đệ Quy

1.cấu trúc dữ liệu ngăn xếp.

Ngăn xếp hoạt động theo nguyên tắc LIFO,vào cuối ra đầu

Trong chương trình tồn tại một bộ nhớ là bộ nhớ ngăn xếp ,cách hoạt động tương tự như cấu trúc dữ liệu ngăn xếp.



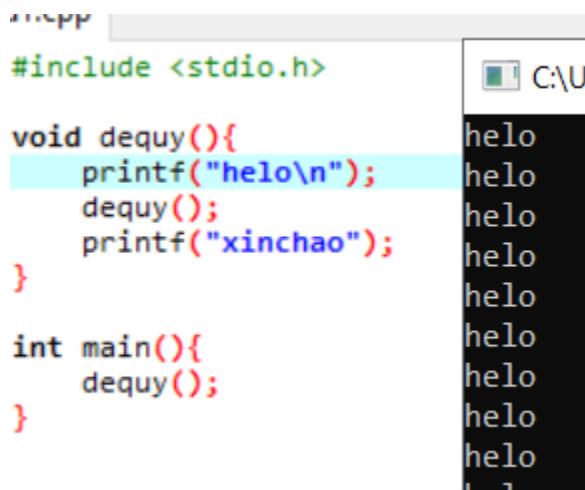
2.Stack frame

Stack frame là một kĩ thuật quản lí bộ nhớ.có nhiệm vụ tạo ra và loại bỏ các biến tạm thời.có thể nói stack frame là tập hợp những thông tin liên quan đếm một chương trình con và chỉ tồn tại trong qua trình chương trình được thực thi.

Mỗi khi một lời gọi hàm được thực hiện stack frame chứa thông tin của hàm đó được đẩy vào bộ nhớ và khi hàm kết thúc thì stack frame được loại khỏi bộ nhớ stack.

3. Hàm Đệ Quy

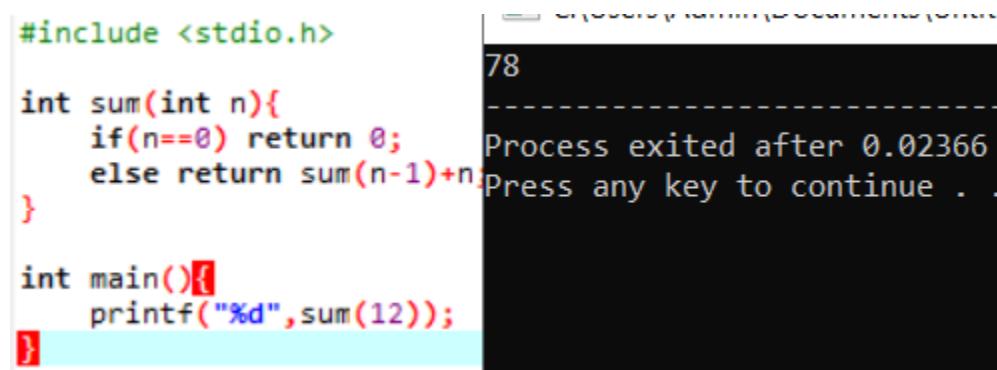
Hàm đệ quy là hàm gọi lại chính nó



```
#include <stdio.h>
void dequy(){
    printf("heloo\n");
    dequy();
    printf("xinchao");
}
int main(){
    dequy();
}
```

sẽ lặp lại vĩnh viễn, sẽ chạy đến lúc tràn bộ nhớ stack.

dùng đệ quy tính tổng các số từ 1→n.



```
#include <stdio.h>
int sum(int n){
    if(n==0) return 0;
    else return sum(n-1)+n;
}
int main(){
    printf("%d",sum(12));
}
```

Đệ quy thường dựa trên công thức toán học gọi là truy hồi và một bài toán con nhỏ nhất. Khi viết đệ quy cần xác định được bài toán con nhỏ nhất làm điểm dừng cho hàm đệ quy và dùng công thức truy hồi để giải bài toán. Không có điểm dừng sẽ bị tràn bộ nhớ stack.

```
#include <stdio.h>

int fibo(int n){
    if(n==2) return 1;
    else if(n==1) return 0;
    else return fibo(n-1)+fibo(n-2);
}
int tohop(int k,int n){
    if(k==0) return 1;
    else return tohop(k-1,n)*(n-k+1)/k;
}
int ucln(int a,int b){
    if(b==0) return a;
    else return ucln(b,a%b);
}
int luythua(int a,int b){
    if(b==0) return 1;
    if(b%2==1) return luythua(a,b/2)*luythua(a,b/2)*a;
    else return luythua(a,b/2)*luythua(a,b/2);
}
int main(){
    int n;
    int k;
    scanf("%d%d",&n,&k);
    printf("%d",luythua(n,k));
}
```

....

*Sàng số nguyên tố O(nloglogn)

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

int n=10000000;
int mang[10000001];

void sang(){
    for(int i=0;i<=n;i++){
        mang[i]=1;
    }
    mang[0]=mang[1]=0;
    for(int i=2;i<=sqrt(n);i++){
        if(mang[i]==1){
            for(int j=i*i;j<=n;j+=i){
                mang[j]=0;
            }
        }
    }
}

int main(){
    sang();
    for(int i=0;i<=1000;i++){
        if(mang[i]==1){
            printf("%d ",i);
        }
    }
}
```

public int __cdecl printf (const char * __restrict__ _Format, ...)

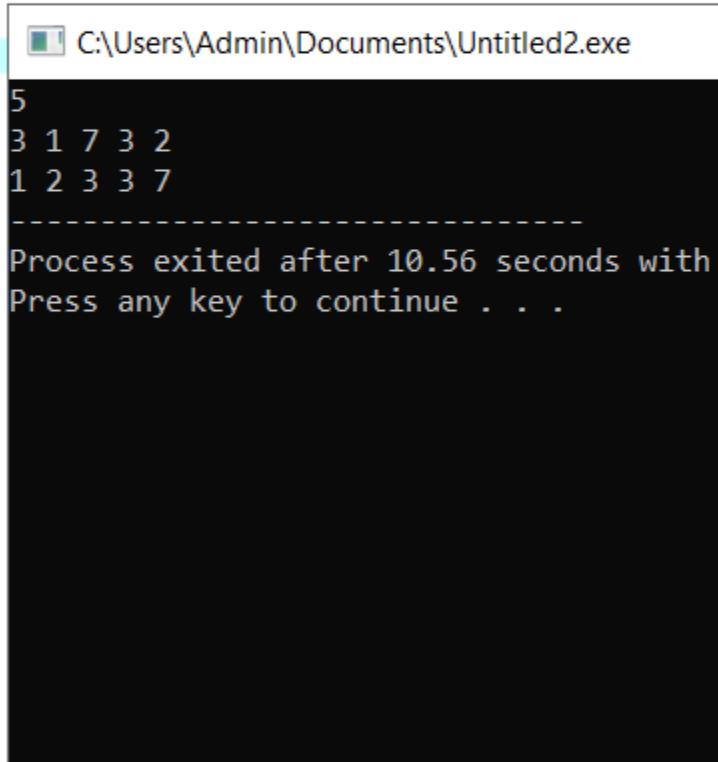
*Selection sort O(n^2)

tìm số nhỏ nhất trong các số chưa được sắp xếp và hoán đổi vị trí với số có vị trí đầu tiên sau
dãy đã được sắp xếp

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

```
void selectionsort(int a[],int n){
    for(int i=0;i<n-1;i++){
        int min=i;
        for(int j=i+1;j<n;j++){
            if(a[j]<a[min]){
                min=j;
            }
        }
        int tmp=a[min];
        a[min]=a[i];
        a[i]=tmp;
    }
}
```

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



```
C:\Users\Admin\Documents\Untitled2.exe
5
3 1 7 3 2
1 2 3 3 7
-----
Process exited after 10.56 seconds with
Press any key to continue . . .
```

*Bubble sort O(n^2)

Đẩy các số lớn hơn về cuối bằng cách liên tục so sánh và hoán vị với số đứng sau

```
#include <stdio.h>

void bubblesort(int a[],int n){
    for(int i=0;i<n-1;i++){
        for(int j=0;j<n-i-1;j++){
            if(a[j]>a[j+1]){
                int tmp = a[j];
                a[j]=a[j+1];
                a[j+1]=tmp;
            }
        }
    }
}

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

*Insertion sort O(n^2)

chèn số vào dãy đã được sắp xếp, so sánh số chưa được chèn vào dãy đã sắp xếp sao cho dãy vẫn tăng dần.

Vòng lặp while có 2 điều kiện cần thoả mãn sẽ liên tục đẩy các số ở vị trí(i-1) lên phía trước và sau vòng lặp giá trị cần chèn sẽ được chèn vào vị trí i-1+1 do tại vị trí i-1 thì điều kiện ko còn thoả mãn.

```
#include <stdio.h>

void insertionsort(int a[],int n){
    for(int i=1;i<n;i++){
        int vitri=i-1;
        int value=a[i];
        while(vitri>=0 && value<a[vitri]){
            a[vitri+1]=a[vitri];
            vitri--;
        }
        a[vitri+1]=value;
    }
}

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

*Counting sort O(n+k)

Đếm mỗi giá trị xuất hiện bao lần rồi in từ bé đến lớn,sử dụng mảng đếm

điều kiện:ko xuất hiện số âm,mảng đếm phải là mảng có kích cỡ K+1 với K là giá trị của phần tử lớn nhất.

```
#include <stdio.h>
#include <math.h>
int cnt[10000001]={0};

void countingsort(int a[],int n){
    int K=-1e9;
    for(int i=0;i<n;i++){
        K=fmax(K,a[i]);
    }
    for(int i=0;i<n;i++){
        cnt[a[i]]++;
    }
    for(int i=0;i<=K;i++){
        int h=cnt[i];
        for(int j=0;j<h;j++){
            printf("%d ",i);
        }
    }
}

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

}
```

*MergeSort O(NlogN)

cần có 1 hàm merge để trộn 2 dãy đã được sắp xếp,với đệ quy ta đã chia được mảng a về mảng chỉ có 1 phần tử và mảng 1 phần tử thì vẫn luôn được sắp xếp.sau đó hàm đệ quy chạy ngược lại và bắt đầu thực hiện hàm merge trong hàm mergesort để sắp xếp từ các phần tử riêng lẻ thành mảng a ban đầu.

```
#include <stdio.h>

void merge(int z[],int l,int r,int n){
    int n1=n-l+1,n2=r-n;
    int y[n1],z[n2];
    for(int i=0;i<n1;i++){
        y[i]=z[l+i];
    }
    for(int i=0;i<n2;i++){
        y[i]=z[n+i];
    }
    int i=0,j=0,cnt=0;
    while(i<n1 && j<n2){
        if(y[i]<y[j]) z[cnt++]=y[i++];
        else z[cnt++]=y[j++];
    }
    while(i<n1){
        z[cnt++]=y[i++];
    }
    while(j<n2){
        z[cnt++]=y[j++];
    }
}

void mergesort(int z[],int l,int r){
    int n=(l+r)/2;
    if(l<r){
        mergesort(z,l,n);
        mergesort(z,n+1,r);
        merge(z,l,r,n);
    }
}

int main(){
    int r;
    scanf("%d",&r);
    int z[r];
    for(int i=0;i<r;i++){
        scanf("%d",&z[i]);
    }
    mergesort(z,0,r-1);
    for(int i=0;i<r;i++){
        printf("%d",z[i]);
    }
}
```

*Quicksort

chọn phần tử cuối cùng và đẩy các phần tử lớn hơn và phía sau,nhỏ hơn về phía trước,sử dụng đệ quy để phân hoạch mảng,đến khi mảng có số phần tử nhỏ hơn thì các số đã tự được sắp xếp tăng dần do luôn nhỏ hơn phần tử được chọn từ đầu hàm quick.

```
#include <stdio.h>

int quick(int a[],int l,int r){
    int i=l-1;
    int pivot=a[r];
    for(int j=l;j<r;j++){
        if(a[j]<=pivot){
            i++;
            int tmp=a[i];
            a[i]=a[j];
            a[j]=tmp;
        }
    }
    i++;
    int tmp=a[r];
    a[r]=a[i];
    a[i]=tmp;
    return i;
}

void quicksort(int a[],int l,int r){
    if(l<r){
        int p=quick(a,l,r);
        quicksort(a,l,p-1);
        quicksort(a,p+1,r);
    }
}
```

*Thuật toán tìm kiếm nhị phân và thuật toán tìm kiếm sắp tuyến tính.

```
#include <stdio.h>

int check(int a[],int n,int x){
    int l=0,r=n-1;
    while(l<=r){
        int mid((l+r)/2);
        if(a[mid]==x){
            return 1;
        }
        else if(a[mid]<x){
            l=mid+1;
        }
        else {
            r=mid-1;
        }
    }
    return 0;
}

int main(){
    int n,x; scanf("%d%d",&n,&x);
    int a[n];
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
    if(check(a,n,x)) printf("yes");
    else printf("no");
}
```

*Biến thể của thuật toán nhị phân.

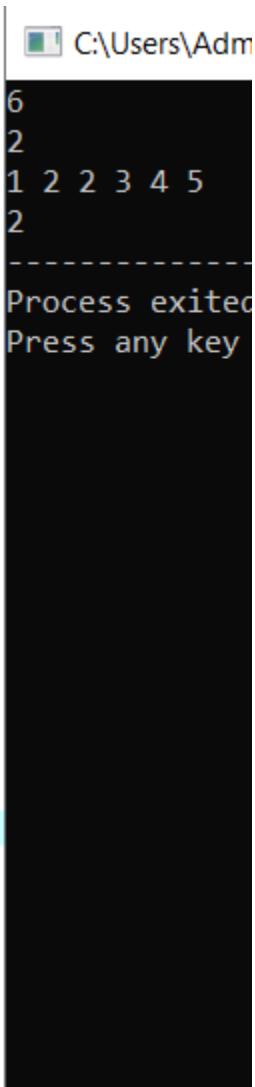
Tìm vị trí đầu tiên của phần tử x

tương tự với tìm phần tử cuối cùng.

```
#include <stdio.h>

int check(int a[], int n, int x){
    int l=0, r=n-1, res=-1;
    while(l<r){
        int m=(l+r)/2;
        if(a[m]<x){
            l=m+1;
        }
        else if(a[m]==x){
            res=m;
            r=m-1;
        }
        else{
            r=m-1;
        }
    }
    return res;
}

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



C:\Users\Admin
6
2
1 2 2 3 4 5
2

Process exited
Press any key

Tìm phần tử đầu tiên trong mảng $\geq x$.

```
#include <stdio.h>

int check(int a[],int n,int x){
    int l=0,r=n-1,res=-1;
    while(l<r){
        int m=(l+r)/2;
        if(a[m]>=x){
            res=m;
            r=m-1;
        }
        else{
            l=m+1;
        }
    }
    return res;
}

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

sàng số nguyên tố

```
#include <stdio.h>
#include <math.h>
int prime[1000001];

void sang(){
    for(int i=0;i<1000001;i++){
        prime[i]=1;
    }
    prime[0]=prime[1]=0;
    for(int i=2;i<=sqrt(1000000);i++){
        if(prime[i]){
            for(int j=i*i;j<1000001;j+=i){
                prime[j]=0;
            }
        }
    }
}

int main(){
    sang();
    int t;scanf("%d",&t);
    while(t--){
        int n;scanf("%d",&n);
        if(prime[n]){
            printf("yes\n");
        }
        else printf("no\n");
    }
}
```

```
C:\Users\/
5
34
no
13
yes
23
yes
14
no
56
no
-----
Process exi
Press any k
```

in ra các số có đúng 3 ước và nhỏ hơn n.

```
#include <stdio.h>
#include <math.h>
int prime[1000001];

void sang(){
    for(int i=0;i<1000001;i++){
        prime[i]=1;
    }
    prime[0]=prime[1]=0;
    for(int i=2;i<=sqrt(1000000);i++){
        if(prime[i]){
            for(int j=i*i;j<1000001;j+=i){
                prime[j]=0;
            }
        }
    }
}

int main(){
    sang();
    int t;scanf("%d",&t);
    while(t--){
        int n;scanf("%d",&n);
        for(int i=2;i<sqrt(n);i++){
            if(prime[i]){
                printf("%lld ",111*i*i);
            }
        }
    }
}
```

Tìm ước số nguyên tố nhỏ nhất

```
#include<stdio.h>
#include<math.h>
int p[100001];

void sang(){
    for(int i=1;i<100001;i++){
        p[i]=i;
    }
    for(int i=2;i<sqrt(100001);i++){
        if(p[i]==i){
            for(int j=i*i;j<100001;
                if(p[j]==j){
                    p[j]=i;
                }
            }
        }
    }
}

int main(){
    int n;scanf("%d",&n);
    sang();
    for(int i=1;i<n;i++){
        printf("%d\n",p[i]);
    }
}
```

đếm tần suất xuất hiện số nguyên tố.

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

int main(){
    int n=0,int tmp;
    int a[10000];
    while(scanf("%d",&tmp)!=-1){
        a[n]=tmp;
        n++;
    }
    for(int i=0;i<n;i++){
        int ck=1;
        for(int j=0;j<i;j++){
            if(a[i]==a[j]){
                ck=0;
                break;
            }
        }
        if(ck){
            if(check(a[i])){
                int cnt=1;
                for(int j=i+1;j<n;j++){
                    if(a[i]==a[j]){
                        cnt++;
                    }
                }
                printf("%d %d",a[i],cnt);
            }
        }
    }
}
```

tìm số tuyén xe buýt ngắn nhất

```
' void quicksort(int a[],int l,int r){  
    if(l<r){  
        int p=quick(a,l,r);  
        quicksort(a,l,p-1);  
        quicksort(a,p+1,r);  
    }  
}
```

```
int main(){  
    int n;int cnt=0;  
    scanf("%d",&n);  
    int a[n];  
    for(int i=0;i<n;i++){  
        scanf("%d",&a[i]);  
    }  
    quicksort(a,0,n-1);  
    for(int i=0;i<n;i++){  
        printf("%d ",a[i]);  
    }  
    int min=1e9+2;  
    for(int i=1;i<n;i++){  
        if(a[i]-a[i-1]<min){  
            min=a[i]-a[i-1];  
            cnt=1;  
        }  
        else if(a[i]-a[i-1]==min){  
            cnt++;  
        }  
    }  
    printf("\n%d %d",min,cnt);  
    return 0;
```

kiểm tra có 3 số nào tạo thành 3 cạnh của tam giác vuông trong mảng

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

    for(int i=0;i<n;i++){
        a[i]=a[i]*a[i];
    }
    quicksort(a,0,n-1);

    for(int i=n-1;i>=2;i--){
        int l=0;
        int r=i-1;
        while(l<r){
            if(a[l]+a[r]>a[i]){
                r--;
            }
            else if(a[l]+a[r]<a[i]){
                l++;
            }
            else{
                printf("yes");
                return 0;
            }
        }
    }
    printf("no");
}
```

*Sử dụng hàm quicksort

nằm trong thư viện “**stdlib.h**”

```
//void qsort(array, so_Luong_phan_tu, kich_thuoc, int (*compar)(const void *, const void*))
```

phần “compar” cách sắp xếp

của hàm qsort có thể là tăng dần giảm dần.....

ví dụ về sắp xếp tăng dần

```
int comp(const void *a, const void *b){  
    int *x=(int*)a;  
    int *y=(int*)b;  
    return *x - *y;  
}  
  
int main(){  
    int n;  
    scanf("%d",&n);  
    int a[n];  
    for(int i=0;i<n;i++){  
        scanf("%d",&a[i]);  
    }  
    qsort(a,n,sizeof(int),comp);  
    for(int i=0;i<n;i++){  
        printf("%d |",a[i]);  
    }  
    public int __cdecl printf (const char * __res  
}
```

sử dụng 2 biến con trỏ a và b để xác định thứ tự của phần tử trong mảng.

cần ép kiểu 2 biến con trỏ theo kiểu dữ liệu được sử dụng trong mảng.

-nếu a và b đã đúng thứ tự mong muốn trả về giá trị âm

-nếu không đúng trả về giá trị dương

*lý thuyết mảng 2 chiều

Cần chỉ ra số hàng số cột của ma trận: hàng trước cột sau.

Nhập vào mảng 2 chiều

```
#include <stdio.h>

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

Duyệt các số nguyên tố trong mảng

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

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

