中山大學
SUN YAT-SEN UNIVERSITY

循环结构

中山大学计算机学院

讲课人：万海

目录

CONTENTS

01 循环结构

02 while循环

03 for循环

04 循环嵌套

05 break/continue

循环结构（Loop）

在日常生活中，我们常常遇到需要重复处理的问题，所以还需要用到**循环结构**来处理这类问题。

求30个整数的和	——>	（重复30次相同的**加法**操作）
统计全班30人的平均成绩	——>	（重复30次相同的**计算**操作）
向系统输入30个人的成绩	——>	（重复30次相同的**输入**操作）
检查30人的成绩是否及格	——>	（重复30次相同的**判断**操作）

针对这一类问题，使用循环结构单一结构用来处理大量重复的操作

while循环

```
while(表达式)
{
    语句块
}
```

```
while(表达式)
    语句
```

在while循环结构下，首先判断**循环条件表达式**是否为真(即给定的条件是否成立)，如果为真，就执行后面的语句/语句块，也称为**循环体。**

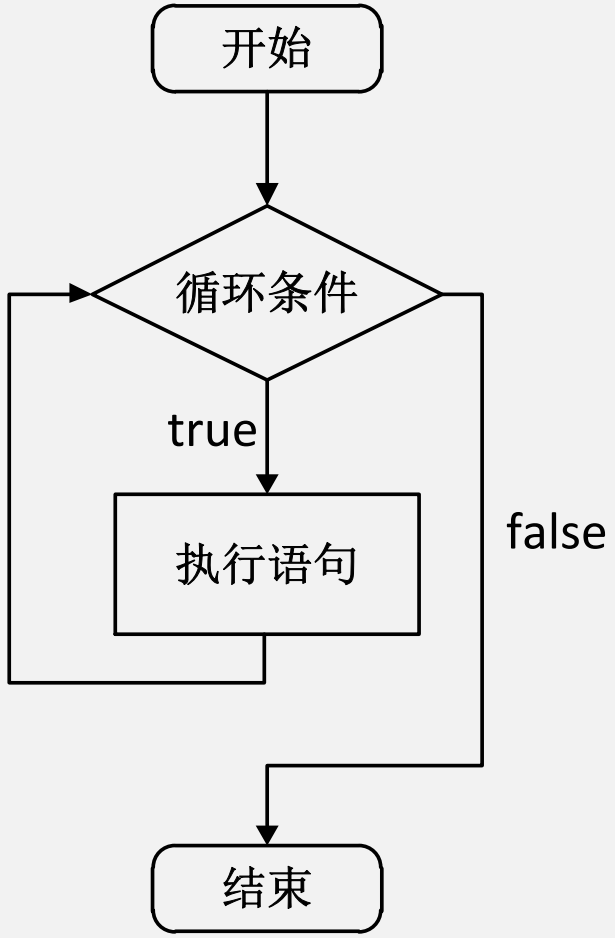
之后不断的重复上述过程，先判断**循环条件表达式**是否为真，为真就执行**循环体。**

直到判断**循环条件表达式**为假，不再执行循环体，离开循环结构。

while循环

```
while(表达式)
{
    语句块
}
```

```
while(表达式)
    语句
```



while循环

```
int i = 0;
while(i < 50)
{
    i++;
    printf("%d\n", i);
}
```

在while循环结构下，可以通过**循环条件表达式**来控制循环体执行的次数。

i < 50就是一个循环条件表达式，通过定义**循环变量**i来控制循环的次数。只要i < 50这个条件成立，该循环结构便会一直执行循环体中的语句。

在循环体中通过语句i++，每执行一次循环会将i的值+1，直到i的值等于50，跳出循环。

输出1,2,3,4,5.....50。　　想想为什么输出50呢？

while循环

```
int i = 1，sum = 0;
while(i <= 100)
{
    sum += i;
    i++;
}
```

求1+2+3+4+...+100？

首先定义变量i的初始值为1，sum的初始值为0。当i > 100时，条件“i <= 100”不假，不在执行循环体中的sum += i，循环结束。

#include <stdio.h> int main(){ while(1){ printf("1"); } return 0; }

while循环

“死循环”

```
#include <stdio.h>
int main(){
    while(1){
        printf("1");
    }
    return 0;
}
```

#include <stdio.h> int main(){ while(1){ printf("1"); } return 0; }

while循环

```
#include <stdio.h>
int main(){
    while(0){
        printf("1");
    }
    return 0;
}
```

#include <stdio.h> int main(){ while(1){ printf("1"); } return 0; }

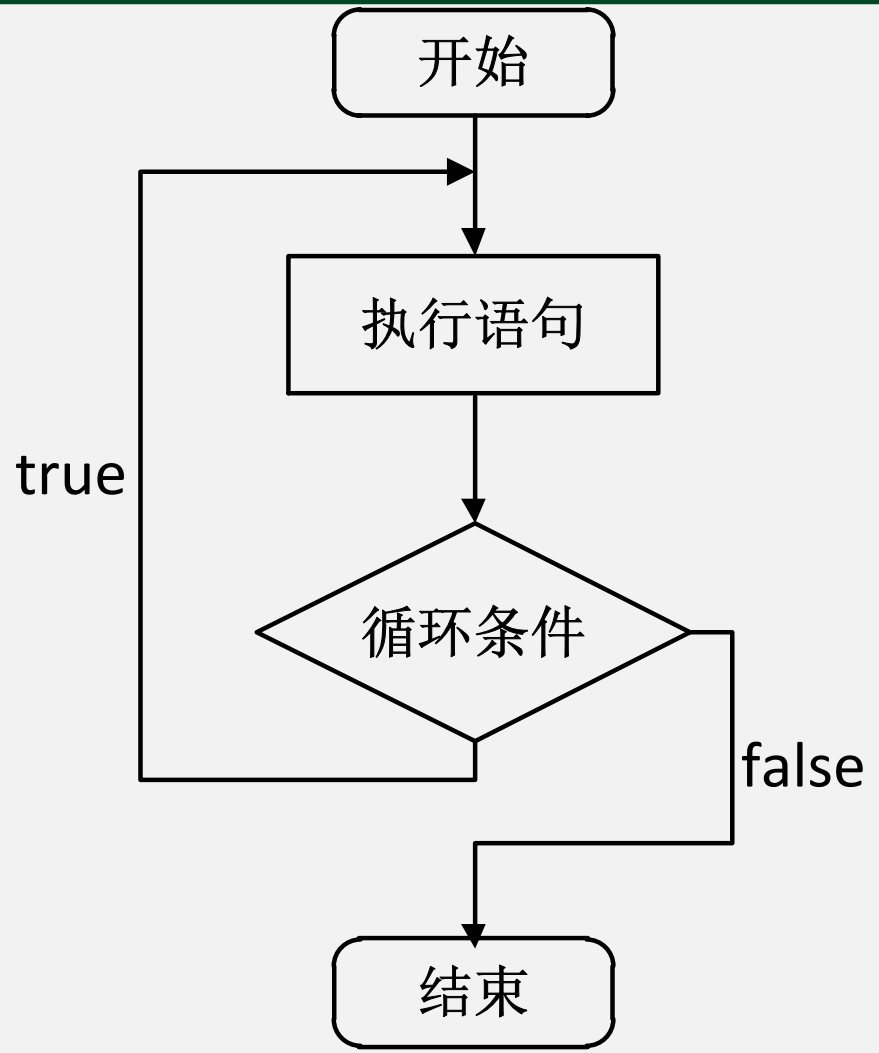
while循环

```
#include <stdio.h>
int main(){
    int n=0;
    printf("Input a string:");
    while(getchar()!='\n') n++;
    printf("Number of characters: %d\n", n);
    return 0;
}
```

do...while循环

```
do
{
      语句块
}
while(表达式);
```

在do...while循环结构下，先执行一次指定的循环体语句，然后判断表达式，当表达式的值为真时，返回重新执行循环体语句，如此反复，直到判断到表达式的值为假，结束循环。



do...while循环

sum的值有什么不同？

```
int i = 11 , sum = 0;
while(i <= 10)
{
    sum += i;
    i++;
}
printf("%d\n", sum);
```

```
int i = 11 , sum = 0;
do
{
    sum += i;
    i++;
}
while(i <= 10);
printf("%d\n", sum);
```

do...while循环

sum的值有什么不同？

至少要执行一次“语句块”

```
int i = 11 , sum = 0;
while(i <= 10)
{
    sum += i;
    i++;
}
printf("%d\n", sum);
```

```
int i = 11 , sum = 0;
do
{
    sum += i;
    i++;
}
while(i <= 10);
printf("%d\n", sum);
```

for循环

```
for(表达式1；表达式2；表达式3）
{
    语句块
}
```

表达式1：

设置初始条件，只执行一次，可以为零个、一个或多个变量设置初始值。

表达式2：

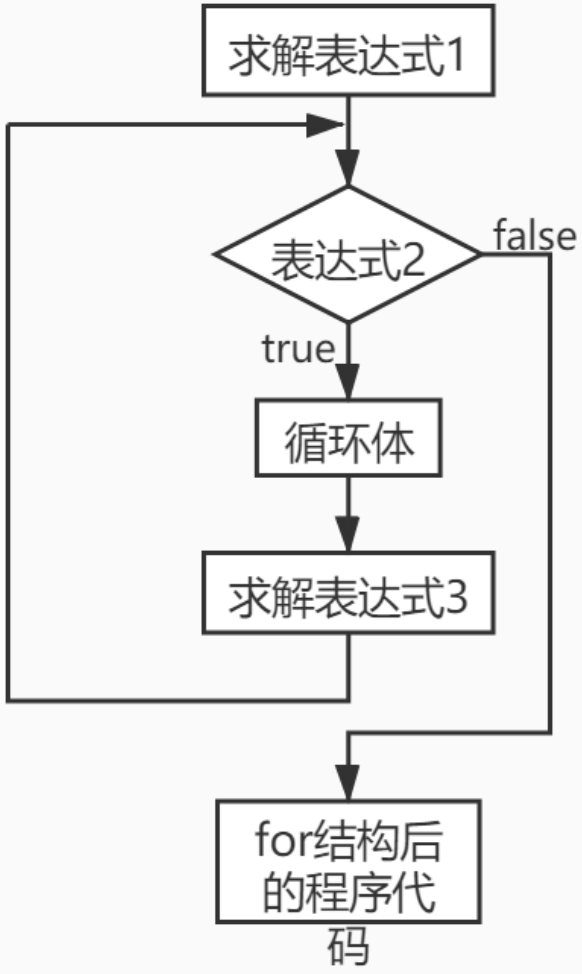
循环条件表达式，根据表达式的真假来决定是否继续执行循环。

表达式3：

作为循环的调整，如使循环变量增加，它是执行完循环体后才进行的。

for循环

```
for(表达式1；表达式2；表达式3）
{
      语句块
}
```



for循环

```
for(循环变量赋初值；循环条件；循环变量增值）
{
      语句块
}
```

最常见的for循环语句形式↑

值得注意的是表达式3，要使循环变量趋于使循环结构”结束”的方向进行，避免死循环

for循环

```
int sum = 0;
for(int i = 1 ; i <= 10 ; i++ )
{
    sum += i;
}
```

等价于

```
int i = 1 , sum = 0;
while(i <= 10)
{
    sum += i;
    i++;
}
```

for循环

```
int sum = 0;
for(int i = 1 ; ; i++ )
{
    sum += i;
}
```

等价于

```
int i = 1 , sum = 0;
while(1)
{
    sum += i;
    i++;
}
```

for循环

```
int sum = 0;
for(int i = 1 ; i <= 10 ; )
{
    sum += i;
    i++ ;
}
```

```
int sum = 0,i = 1;
for( ; i <= 10 ; )
{
    sum += i;
    i++ ;
}
```

等价于

```
int i = 1 , sum = 0;
while(i <= 10)
{
    sum += i;
    i++;
}
```

for循环

3个表达式可以同时省略

```
for( ; ; )

while(1)
```

for(i=0,j=100; i<=100; i++,j--) k=i+j;

for循环

“表达式1”和“表达式3”可以是一个简单表达式也可以是逗号表达式

```
for( sum=0,i=1; i<=100; i++ )  sum=sum+i;
```

```
for( i=0,j=100; i<=100; i++,j-- )  k=i+j;
```

#include <stdio.h> int main(){ while(1){ printf("1"); } return 0; }

while循环

```
#include <stdio.h>
int main(){
    int n=0;
    printf("Input a string:");
    while(getchar()!='\n') n++;
    printf("Number of characters: %d\n", n);
    return 0;
}
```

#include <stdio.h> int main(){ while(1){ printf("1"); } return 0; }

while循环

```
#include <stdio.h>
int main(){
    int n=0;
    printf("Input a string:");
    \\while(getchar()!='\n') n++;
    for( n=0; (getchar())!='\n'; n+=1 );
    printf("Number of characters: %d\n", n);
    return 0;
}
```

循环嵌套

```
while(表达式1)
{
    while(表达式2)
    {...}
}
```

```
for(表达式1.1;表达式1.2；表达式1.3)
{
    for(表达式2.1;表达式2.2；表达式2.3)
    {...}
}
```

一个循环体内又包含另一个完整的循环结构，称为**循环的嵌套**

循环嵌套（ Nest ）

```
for(int i = 1;i <= 2; i++)
{
    for(int j = 1;j <= 3; j++)
        printf("i = %d,j = %d, i*j = %d\n",i,j,i*j);
}
```

输出结果:

```
i = 1,j = 1,i*j = 1
i = 1,j = 2,i*j = 2
i = 1,j = 3,i*j = 3
i = 2,j = 1,i*j = 2
i = 2,j = 2,i*j = 4
i = 2,j = 3,i*j = 6
```

在该结构中，首先进入第一层循环i = 1，满足循环条件i <= 2，执行循环体的内容，而循环体是一个单独的循环，于是执行这个单独的循环，当执行完单独的循环，在执行i++

循环嵌套（ Nest ）

输出一个4×4的整数矩阵

```
1    2    3    4
2    4    6    8
3    6    9    12
4    8    12   16
```

循环嵌套（ Nest ）

输出一个4×4的整数矩阵

```
1   2   3   4
2   4   6   8
3   6   9   12
4   8   12  16
```

```
#include <stdio.h>
int main()
{
    int i, j;
    for(i=1; i<=4; i++){  //外层for循环
        for(j=1; j<=4; j++){  //内层for循环
            printf("%-4d", i*j);
        }
        printf("\n");
    }
    return 0;
}
```

循环嵌套（ Nest ）

输出九九乘法表

```
1*1=1
2*1=2   2*2=4
3*1=3   3*2=6   3*3=9
4*1=4   4*2=8   4*3=12  4*4=16
5*1=5   5*2=10  5*3=15  5*4=20  5*5=25
6*1=6   6*2=12  6*3=18  6*4=24  6*5=30  6*6=36
7*1=7   7*2=14  7*3=21  7*4=28  7*5=35  7*6=42  7*7=49
8*1=8   8*2=16  8*3=24  8*4=32  8*5=40  8*6=48  8*7=56  8*8=64
9*1=9   9*2=18  9*3=27  9*4=36  9*5=45  9*6=54  9*7=63  9*8=72  9*9=81
```

循环嵌套（ Nest ）

输出九九乘法表

```
#include <stdio.h>
int main(){
    int i, j;
    for(i=1; i<=9; i++){  //外层for循环
        for(j=1; j<=i; j++){  //内层for循环
            printf("%d*%d=%-2d  ", i, j, i*j);
        }
        printf("\n");
    }

    return 0;
}
```

break/continue语句

```
break;
```

break语句可以使当前流程跳出switch结构，继续执行switch结构后的下一条语句。

break语句还用来从循环体中跳出整个循环结构，即提前结束循环，执行循环结构后的下一条语句。

```
continue;
```

有时候并不希望终止整个循环，只是想提前结束本次循环，而接着执行下次循环

break/continue语句

```
for(int i = 1;i <= 10; i++)
{
    if(i == 5) break;
    printf("i = %d\n",i);
}
```

```
i = 1
i = 2
i = 3
i = 4
i = 5
```

这个for循环结构的作用为打印数字1-10

第一个程序中：

当i = 5时，满足if语句的判断，执行break语句，此后跳出整个循环，循环结束。

```
for(int i = 1;i <= 10; i++)
{
    if(i == 5) continue;
    printf("i = %d\n",i);
}
```

```
i = 1
i = 2
i = 3
i = 4
i = 6
i = 7
i = 8
i = 9
i = 10
```

第二个程序中：

当i = 5时，满足if语句的判断，执行continue语句，此后跳过本次循环中continue语句后面语句的执行，转入下一次循环。

二元一次方程组求解(loop1, loop1_1)

给定以下的二元一次方程组，求一个整数解x_1, x_2，要求$-100 \leq x_1, x_2 \leq 100$.
若无解则输出-1, 若有多解，则输出x_1较小的解，若x_1相同，则输出x_2较小的解。

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases}$$

输入格式

一行六个整数，分别表示$a_{11}, a_{12}, b_1, a_{21}, a_{22}, b_2$.六个整数的绝对值不超出$10^5$

输出格式

一行两个整数，分别表示x_1, x_2

输入样例

1 1 3 1 2 5

输出样例

1 2

```
#include <stdio.h>
int main()
{
    int a11, a12, b1, a21, a22, b2;
    scanf("%d%d%d%d%d%d", &a11, &a12, &b1, &a21, &a22, &b2);

    int flag=0; //判断是否有解
    for (int x1=-100; x1<=100; ++x1)
        for (int x2=-100; x2<=100; ++x2)
            if (a11*x1+a12*x2==b1 && a21*x1+a22*x2==b2)
            {
                flag=1;
                printf("%d %d\n", x1, x2);
                break;
            }
    if (!flag) printf("-1");
    return 0;
}
```

```
#include <stdio.h>
int main()
{
    int a11, a12, b1, a21, a22, b2;
    scanf("%d%d%d%d%d%d", &a11, &a12, &b1,
&a21, &a22, &b2);
    int flag=0; //判断是否有解
    int x1=-100, x2;
    while (x1<=100)
    {
        x2=-100;
        while (x2<=100)
        {
            if (a11*x1+a12*x2==b1 &&
a21*x1+a22*x2==b2)
            {
                flag=1;
                printf("%d %d\n", x1, x2);
                break;
            }
                x2+=1;
        }
        if (flag) break;
        x1+=1;
    }
    if (!flag) printf("-1");
    return 0;
}
```

质数计数(loop2)

给定一个数n，求2~n有多少个质数

输入格式

一行一个整数n，$2 \leqslant n \leqslant 1000$

输出格式

一行一个整数，表示质数数目

输入样例

10

输出样例

4

```
#include <stdio.h>
int main()
{
    int n, ans=0;
    scanf("%d", &n);
    for (int i=2; i<=n; ++i)
    {
        int flag=1;
        for (int j=2; j*j<=i; ++j)
            if (i%j==0)
            {
                flag=0;
                break;
            }
        if (flag) ans++;
    }
    printf("%d\n", ans);
    return 0;
}
```

倍数输出(loop3)

给定一个数n，输出n行，第i行从小到大输出i的倍数（不超过n）

输入格式

一行一个整数n，$1 \leqslant n \leqslant 100$

输出格式

如题所示

输入样例

5

输出样例

1 2 3 4 5

2 4

3

4

5

```
#include <stdio.h>

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

中山大學
SUN YAT-SEN UNIVERSITY

期中复习

大小写转换(upper_lower_case)

输入一个字符串，将字符串中的大写字母改为对应的小写字母，将小写字母改为对应的大写字母，其它字符保持不变。
输入格式
一行一个字符串
输出格式
一行一个字符串
输入样例
ab#CD
输出样例
AB#cd

```c
#include <stdio.h>

int main()
{
    char ch;
    while (scanf("%c", &ch)!='\n' && ch!=EOF)
        if (ch>='a' && ch<='z') printf("%c", ch-'a'+'A');
        else if (ch>='A' && ch<='Z') printf("%c", ch-'A'+'a');
        else printf("%c", ch);
    return 0;
}
```

字符串输出(string)

输出

"\('o')/"

```
#include <stdio.h>

int main()
{
    printf("\"\\(\'o\')/\"");
    return 0;
}
```

输出三角形(tri)

输入一个数$n \geqslant 3$，输出一个三角形

输入样例1

3

输出样例1

见右

输入样例2

4

输出样例2

见右

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

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

```
#include <stdio.h>
int main()
{
    int n;
    scanf("%d", &n);
    for (int i=1; i<n; ++i) printf(" ");
    printf("*");
    printf("\n");
    for (int i=2; i<n; ++i)
    {
        for (int j=1; j<=n-i; ++j) printf(" ");
        printf("*");
        for (int j=1; j<=i*2-3; ++j) printf(" ");
        printf("*");
        printf("\n");
    }
    for (int i=1; i<=n*2-1; ++i) printf("*");
    printf("\n");
    return 0;
}
```

一元二次方程求解(equ)

给定一元二次方程$ax^2+bx+c=0$的系数a, b, c， 求方程的较大的解，无解或无穷解输出-1.

输出格式

一行三个整数a, b, c

输出格式

一行一个浮点数表示方程的较大解（保留两位小数）， 或输出-1

输入样例

1 -8 16

输出样例

4.00

思路

如果这是一道高考数学题，你会怎么作答？

解：

（1）若$a=0$且$b=0$, 则化简为$c=0$，无解或无穷解

（2）若$a=0$且$b\neq 0$, 则方程的解为$x=-c/b$

（3）若$a\neq 0$

1. 若$\Delta=b^2-4ac<0$，则无解
2. 若$\Delta\geq 0$,

若$a>0$，则较大解为$\frac{-b+\Delta}{2a}$

若$a<0$，则较大解为$\frac{-b-\Delta}{2a}$

```c
#include <stdio.h>
#include <math.h>
int main()
{
    int a, b, c;
    scanf("%d%d%d", &a, &b, &c);
    if (a==0 && b==0) printf("-1\n"); //方程化简为c=0
    else if (a==0) printf("%.2lf\n", -(double)c/b); //方程化简为bx+c=0
    else if (b*b-4*a*c<0) printf("-1\n"); //只有二次方程才能用这个条件判断无解
    else
    {
        double sdelta=sqrt(b*b-4*a*c), ans;
        //方案1
        if (a>0) ans=(-b+sdelta)/a/2;
        else ans=(-b-sdelta)/a/2;
        printf("%.2lf\n", ans);
    }
    return 0;
}
```

注意判断条件的顺序

```c
//方案2
double x1=(-b+sdelta)/a/2, x2=(-b-sdelta)/a/2;
ans=(x1>x2? x1:x2);
```

阶乘之和(frac)

输入一个数n，输出$1!+2!+\cdots+n!$

输入样例

3

输出样例

9

```
#include <stdio.h>
typedef long long LL;

int main()
{
    int n;
    scanf("%d", &n);
    LL ans=0;

    for (int i=1; i<=n; ++i)
    {
        LL frac=1;
        for (int j=1; j<=i; ++j) frac*=j;
        ans+=frac;
    }
    printf("%lld\n", ans);
    return 0;
}
```

中山大學
SUN YAT-SEN UNIVERSITY

身份证号码校验(id)

身份证一共18位，将前17分别乘以不同的系数，然后相加，得到的结果除以11，得到的结果便是第18位（X代表10）

输入格式

一共18行，每一行两个数，分别表示身份证第i位以及对应系数

第18行的系数为-1

输出格式

正确则输出YES，错误则输出NO

输入样例

见右

输出样例

YES

```
0 7
1 9
2 10
3 5
4 8
5 4
6 2
7 1
8 6
9 3
0 7
1 9
2 10
3 5
4 8
5 4
6 2
1 -1
```

```
#include <stdio.h>

int main()
{
    char ch;
    int x, s=0;
    for (int i=1; i<=18; ++i)
    {
        scanf("%c%d", &ch, &x);
        if (ch=='X') s+=10*x;
        else s+=(ch-'0')*x;
        if (i!=18) scanf("\n");
    }
    if (s%11==0) printf("YES\n");
    else printf("NO\n");
    return 0;
}
```

水仙花数(flower)

水仙花数是指一个三位数，它的每个位上的数字的三次幂之和等于它本身，求所有水仙花数。

$153 = 1^3 + 5^3 + 3^3$是水仙花数

```
#include <stdio.h>

int main()
{
    for (int num=100; num<1000; ++num)
    {
        int x1=num/100, x2=num/10%10, x3=num%10;
        if (x1*x1*x1+x2*x2*x2+x3*x3*x3==num) printf("%d\n", num);
    }
    return 0;
}
```

回文数(palin)

若一个数正读和反读相同，则称为回文数，如121，1221；123则不是
问题一：输入一个数n，判断是否是回文数
问题二：输入一个数n，判断1~n有多少个回文数

```c
#include <stdio.h>

int main()
{
    int n;
    scanf("%d", &n);
    int num=n, reverse=0;
    while (num!=0)
    {
        reverse=reverse*10+num%10;
        num/=10;
    }
    if (n==reverse) printf("YES");
    else printf("NO");
    return 0;
}
```

```c
#include <stdio.h>
int main()
{
    int n, ans=0;
    scanf("%d", &n);
    for (int i=1; i<=n; ++i)
    {
        int num=i, reverse=0;
        while (num!=0)
        {
            reverse=reverse*10+num%10;
            num/=10;
        }
        if (i==reverse) ans++;
    }
    printf("%d\n", ans);
    return 0;
}
```

天数(day)

给定某年某月某日，输出这一天是这一年的第几天

输入样例

2021 10 25

输出样例

298

```c
#include <stdio.h>
int main()
{
    int year, month, day;
    int ans=0;
    scanf("%d%d%d", &year, &month, &day);

    for (int i=1; i<month; ++i)
    {
        switch (i)
        {
            case 1:
            case 3:
            case 5:
            case 7:
            case 8:
            case 10:
            case 12:
                ans+=31;
                break;
            case 2:
                if (year%400==0 ||
(year%100!=0 && year%4==0)) ans+=29;
                else ans+=28;
                break;
            default:
                ans+=30;
        }
    }

    ans+=day;
    printf("%d\n", ans);

    return 0;
}
```