



# **CS10003:** **Programming & Data Structures**

**Dept. of Computer Science & Engineering**  
**Indian Institute of Technology Kharagpur**

*Autumn 2020*



# Iterations and Loops



# Loops

Group of statements that are executed repeatedly while some condition remains true

Each execution of the group of statements is called an **iteration** of the loop



# The Essentials of Repetition

## Loop

Group of instructions computer executes repeatedly while some condition remains true

## Counter-controlled repetition

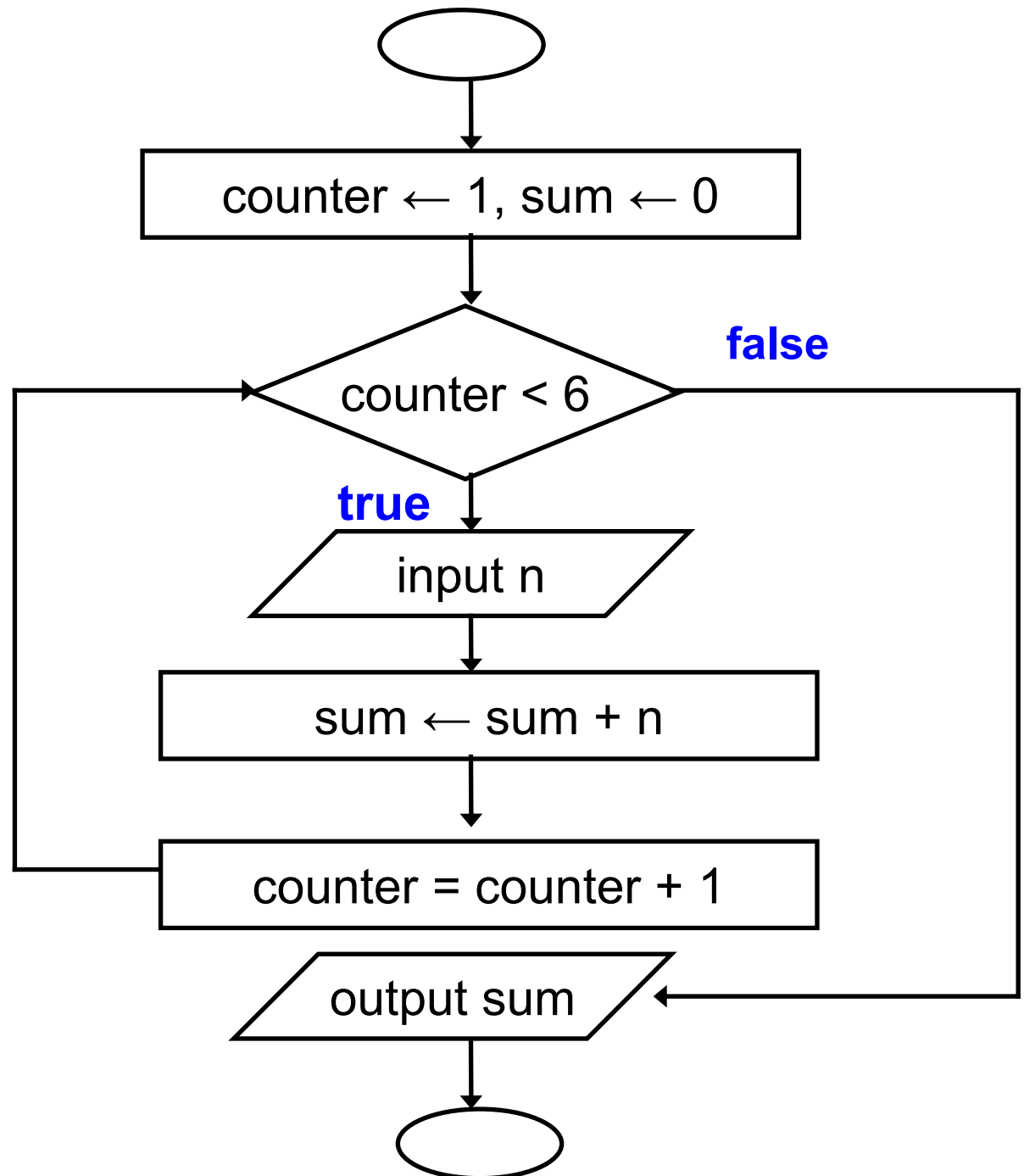
Definite repetition - know how many times loop will execute  
Control variable used to count repetitions

## Sentinel-controlled repetition

Indefinite repetition  
Used when number of repetitions not known  
Sentinel value indicates "end of data"

# Example

Read 5 integers  
and display the  
their sum

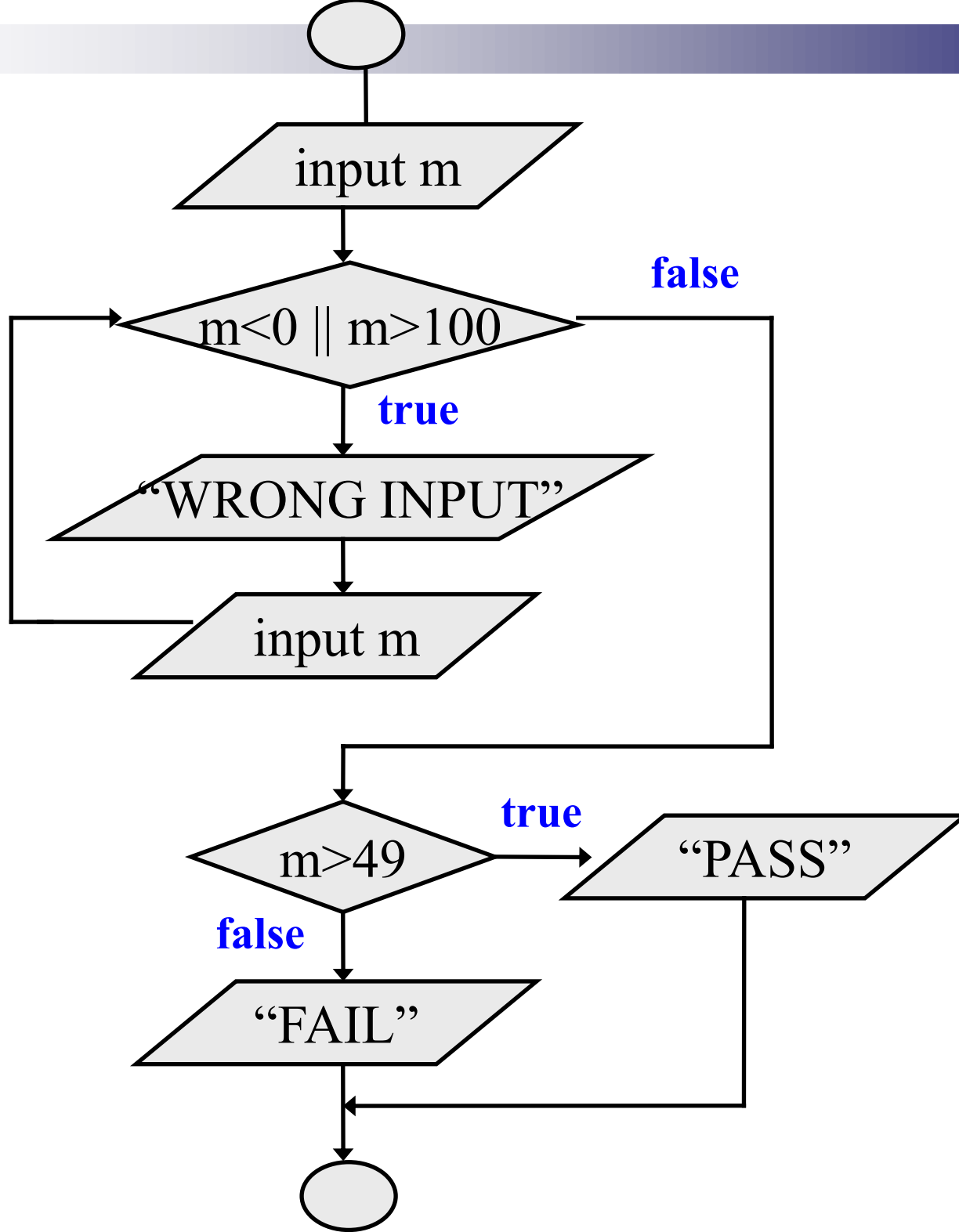


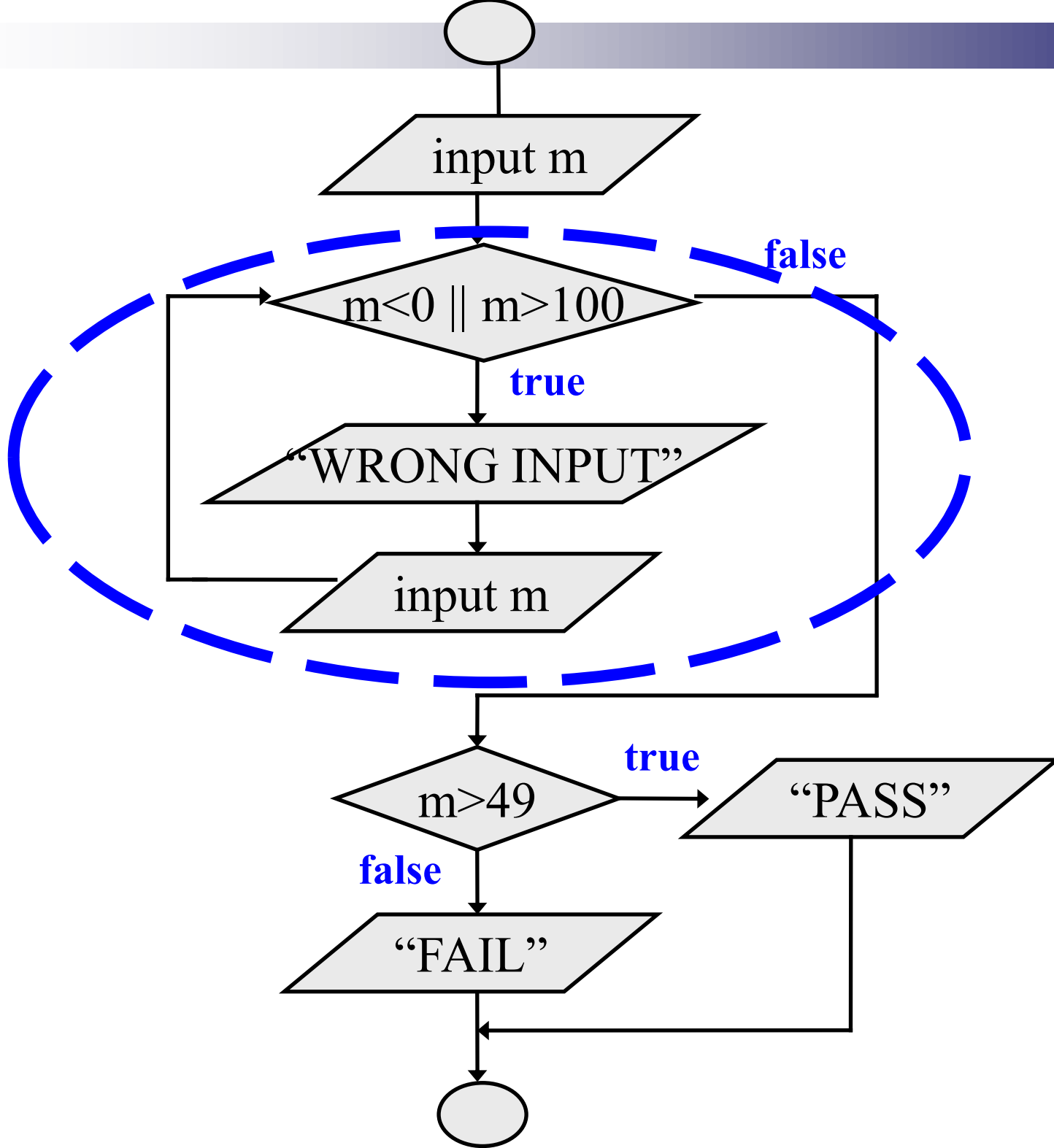


# Example

Given an exam marks as input, display the appropriate message based on the rules below:

- If marks is greater than 49, display “PASS”, otherwise display “FAIL”
- However, for input outside the 0-100 range, display “WRONG INPUT” and prompt the user to input again until a valid input is entered







# Looping: **while** statement

```
while (expression)  
    statement;
```

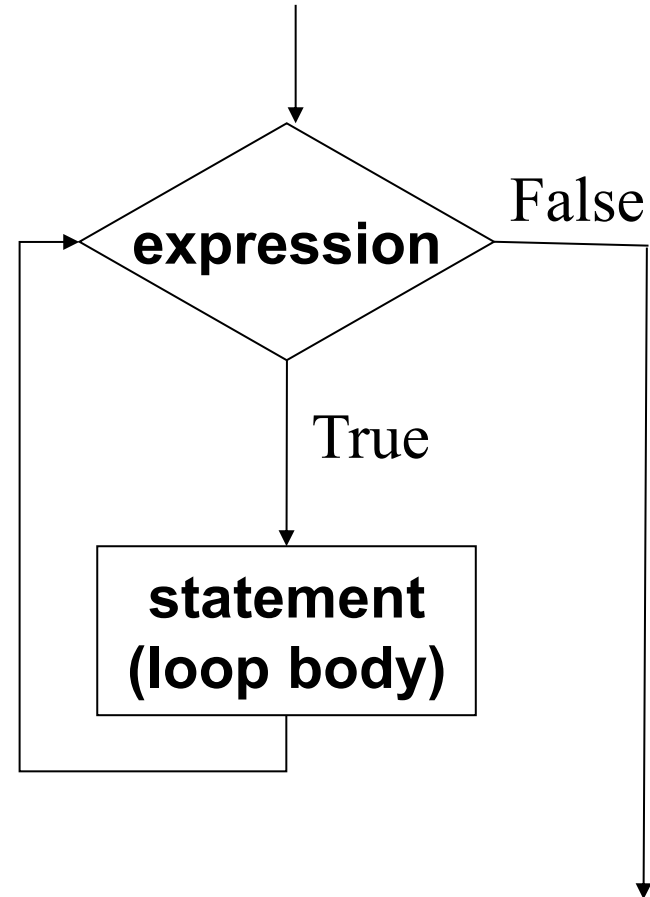
```
while (expression) {  
    Block of statements;  
}
```

The condition to be tested is any expression enclosed in parentheses. The expression is evaluated, and if its value is non-zero, the statement is executed. Then the expression is evaluated again and the same thing repeats. The loop **terminates** when the expression evaluates to 0.

# Looping: **while** statement

```
while (expression)  
    statement;
```

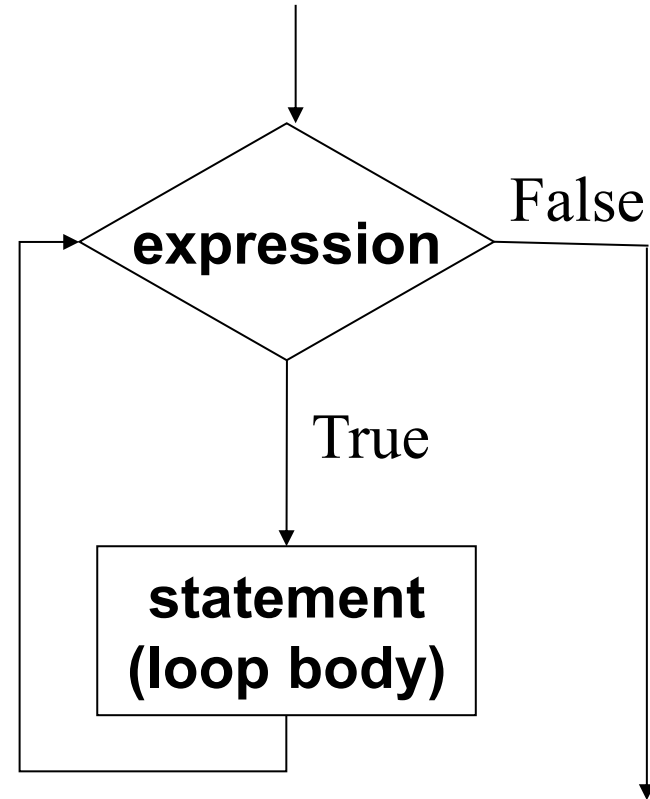
```
while (expression) {  
    Block of statements;  
}
```



# Looping: **while** statement

```
while (expression)  
    statement;
```

```
while (expression) {  
    Block of statements;  
}
```



The condition to be tested is any expression enclosed in parentheses. The expression is evaluated, and if its value is non-zero, the statement is executed. Then the expression is evaluated again and the same thing repeats. The loop **terminates** when the expression evaluates to 0.

# Example

```
int i = 1, n;  
scanf("%d", &n);  
while (i <= n) {  
    printf ("Line no : %d\n",i);  
    i = i + 1;  
}
```



# Example

```
int weight;  
scanf("%d", &weight);  
while ( weight > 65 ) {  
    printf ("Go, exercise, ");  
    printf ("then come back. \n");  
    printf ("Enter your weight: ");  
    scanf ("%d", &weight);  
}
```

# Sum of first N natural numbers

```
int main() {  
    int N, count, sum;  
    scanf ("%d", &N) ;  
    sum = 0;  
    count = 1;  
    while (count <= N) {  
        sum = sum + count;  
        count = count + 1;  
    }  
    printf ("Sum = %d\n", sum) ;  
    return 0;  
}
```


$$\text{SUM} = 1^2 + 2^2 + 3^2 + \dots + N^2$$

```
int main() {  
    int N, count, sum;  
    scanf ("%d", &N) ;  
    sum = 0;  
    count = 1;  
    while (count <= N) {  
        sum = sum + count * count;  
        count = count + 1;  
    }  
    printf ("Sum = %d\n", sum) ;  
    return 0;  
}
```

# Compute GCD of two numbers

```
int main() {  
    int A, B, temp;  
    scanf ("%d %d", &A, &B);  
    if (A > B) {  
        temp = A; A = B; B = temp;  
    }  
    while ((B % A) != 0) {  
        temp = B % A;  
        B = A;  
        A = temp;  
    }  
    printf ("The GCD is %d", A);  
    return 0;  
}
```

```
12 ) 45 ( 3  
    36  
    9 ) 12 ( 1  
        9  
        3 ) 9 ( 3  
            9  
            0
```

**Initial:        A=12, B=45**  
**Iteration 1: temp=9, B=12, A=9**  
**Iteration 2: temp=3, B=9, A=3**  
**B % A = 0    → GCD is 3**



# Double your money

Suppose your Rs 10000 is earning interest at 1% per month. How many months until you double your money ?

```
int main() {
    double my_money = 10000.0;
    int n=0;
    while (my_money < 20000.0) {
        my_money = my_money * 1.01;
        n++;
    }
    printf ("My money will double in %d months.\n",n);
    return 0;
}
```

# Maximum of positive Numbers

```
int main() {  
    double max = 0.0, next;  
    printf ("Enter positive numbers, end with 0 or a  
negative number\n");  
    scanf("%lf", &next);  
    while (next > 0) {  
        if (next > max) max = next;  
        scanf("%lf", &next);  
    }  
    printf ("The maximum number is %lf\n", max) ;  
    return 0;  
}
```

# Find the sum of digits of a number

```
int main()
{
    int n, sum=0;
    scanf ("%d", &n);
    while (n != 0) {
        sum = sum + (n % 10);
        n = n / 10;
    }
    printf ("The sum of digits of the number is %d \n", sum);
    return 0;
}
```



**Thank You!**