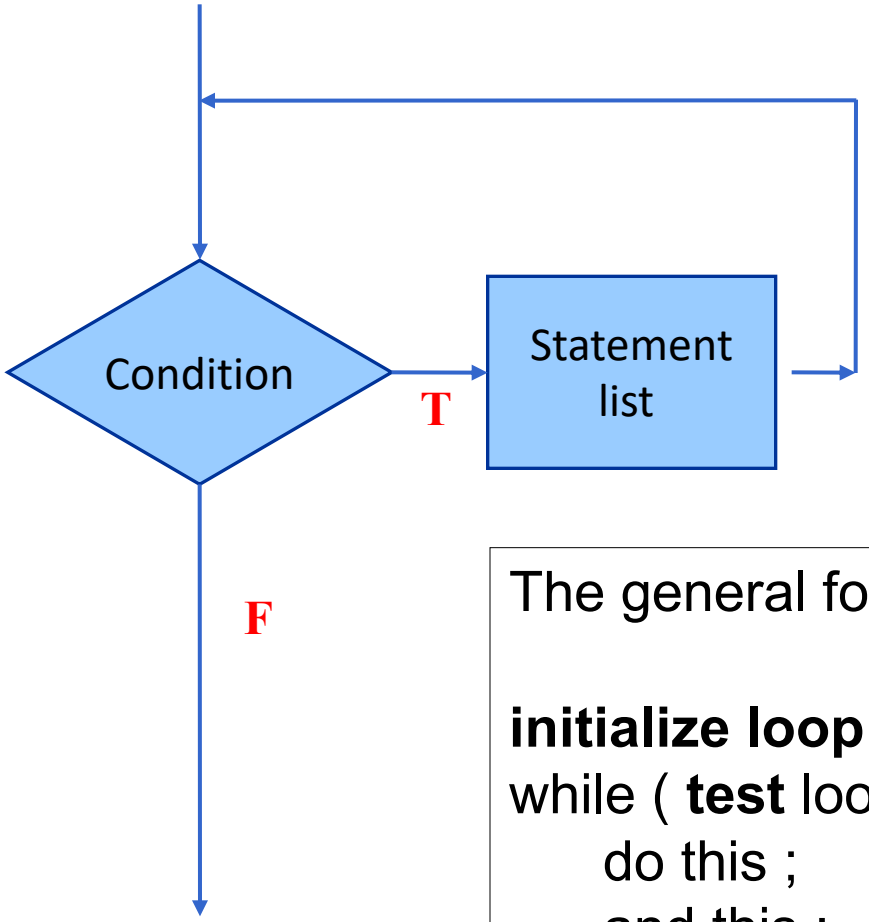
while
and
do..while
Loops

The while loop




```
while(Condition){
   // body of loop
    Statement(s)
}
```

The general form of **while is as shown below:**

initialize loop counter ;
while (**test** loop counter using **a condition**) {
    do this ;
    and this ;
    increment/decrement loop counter ;
}

The while Loop

- Here, **statement(s)** may be a single statement or a block of statements.
- The **condition** may be any expression, and true is any nonzero value. The loop iterates while the condition is true.
- When the condition becomes false, the program control passes to the line immediately following the loop.
- The condition of the loop is tested before the body of the loop is executed, hence it is called an entry-controlled loop.
- *Note: If while loop condition is never false then loop become infinite loop.*

Comparison of ‘for’ and ‘while’ loop

for loop

```
#include<stdio.h>
int main () {
      int n;
      for(n = 1; n <=5; n++) {
            printf("C while loops: %d\n", n);
       }
      return 0;
}
```

while loop

```
#include<stdio.h>
int main () {
      int n = 1;
      while( n <= 5 )    {
            printf("C while loops: %d\n", n);
            n++;
      }
   return 0;
}
```

Both are ENTRY CONTROL Loops

Calculation of simple interest for 3 sets of p, n and r

```
void main ( ) {
   int p, t, count ;
   float r, si ;
   count=1;
   while(count <= 3)          {
            printf ( "Enter values of p, t, and r " ) ;
            scanf ( "%d %d %f", &p, &t, &r ) ;
            si = (p * n * r ) / 100 ;
            printf ( "Simple Interest = Rs.%f\n", si ) ;
            count=count+1; //count++;
   }
}
```

- In the place of condition there can be any other valid expression which evaluates to a non-zero value

1. **while(1) { ……}**
2. **while(-5) { ……}**
3. **while(0) { ……..}**
4. **while (i + j*2) { ……}**
5. **while (i >= 10 && j <= 15) {……..}**

- Must test a condition that will **eventually become false**

```
int i = 1 ;
while ( i <= 10 )
    printf ( "%d\n", i ) ;
```

- We can also use decrement

```
int i = 5 ;
while (i>= 1 )
{
    printf ("\nMake the computer literate!" ) ;
    i = i - 1 ;
}
```

- Loop counter can be float also.

```
float a = 10.0 ;
while (a < 10.5 ) {
    printf ("\n that’s execute!" ) ;
    a=a+0.1;
}
```

Example 1: while

```
char ans = 'n';

while (ans != 'Y' && ans != 'y'){

   printf("Will you pass this exam?");
   scanf("%c\n",&ans);
}

printf("Great!!");
```

Can I put ; here?

No!!

Example 2: `while`

```
int a, b, sum;
printf("This program will return the ");
printf("summation of integers from a to b.\n\n");
printf("Input two integers a and b:");
printf("%d%d", &a, &b);
        sum = 0;
while (a <= b) {
    sum += a;
    a++;
}
printf("The sum is %d", sum);
```

Don't forget to set `sum = 0;`

`sum = sum + a;`

+= ,*=,-=,/=,%=

compound assignment operator

They modify the value of the operand to the left of them.

Example 3: `while`

```
int a, b, sum=0;
printf(“This program will return the sum ”);
printf( “of odd numbers between a and b.\n\n”);
printf(“Input two integers a and b:”);
scanf(“%d%d”, &a ,&b);

while (a <= b) {
  if (a % 2)
     sum += a;
  a++;
}

printf(“The answer is %d”,sum);
```

Counter Controlled loop

```
int i = 1 ;
printf(“How many times repeat these statements”);
scanf(“%d”,&n)
while (i<= 10 )  { //while i is less than or equal to n

   printf ("\nWelcome!" ) ;
   i = i+1 ;
}
```

Try following now:

1 Print numbers from 10 to 1

2 Print numbers from n to m. n and m are accepted from user.

3 Print all numbers divisible by ‘n’ , between 1 to 1000. ‘n’ accepted from user

4 Print sum of numbers from 1 to 10

5. Print Alphabets from A – Z, a- z, z-a, Z-A

6.ASCII Values of characters ‘0’ – ‘9’

usage of for /while

The 'for' loop used when we already knew the number of iterations

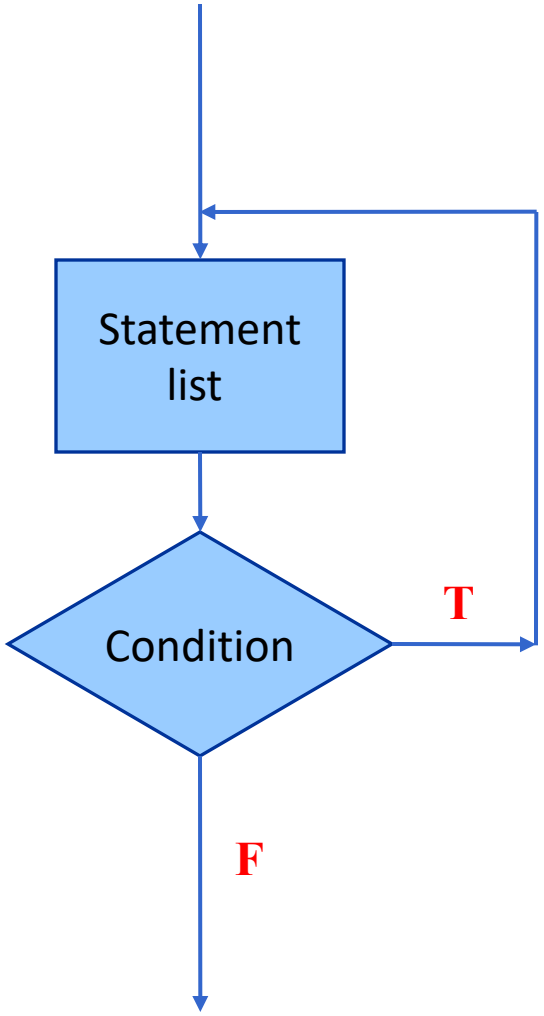
The 'while' loop used when the number of iteration are not exactly known.

HW: **Read more about features of for and while loop**

Condition Controlled loop

```
char ans = 'y';

while (ans == 'y' || ans == 'Y') {
 printf("Enter students details");
    .......
 Printf("Do you want to submit another student
 details");
 scanf("%c",&ans);
}
....
....
printf("Great!!");
```

do-while




```
do {
   printf("Will you pass this
   exam?");
    scanf("%c",&ans);
} while (ans != "Y");

printf("Great!!");
```

```
do {
   Statement list
} while (Condition);
```

; is required

do-while loop

- A do-while loop is similar to a while loop, except that a do-while loop is execute at **least one time.**
- A do while loop is a control flow statement that executes a block of code at least once, and then repeatedly executes the block, or not, depending on a given condition at the end of the block (in while).
- **It is an exit-controlled loop.**

Example 1: do-while

```
#include<stdio.h>

int main () {
  //variable Initialization
  int n = 1,times=5;

  /* do loops execution */
  do    {
     printf("C do while loops: %d\n", n);
     n = n + 1;
  }
  while( n <= times );
  return 0;
}
```

Example 2: do-while

```
int i;

....

do {

 printf( “Please input a number between 10
  and 20:”);
 scanf(“%d”, &i);
 printf(“%d”,i);
} while (i < 10 || i > 20);

......
```

Example 3: do-while

- Suppose your Rs 10000 is earning interest at 1% per month. How many months required to double your money?

```
float my_money=10000.0;
int n=0;
do {
     my_money = my_money*1.01;
     n++;
} while (my_money < 20000.0)

printf (“My money will double in %d months.\n”,n);
```

Nested loops (loop in a loop)

The syntax for a nested for loop statement in C is as follows −

```
for (init; condition; increment) {
   for (init; condition; increment) {
     statement(s);
   }
   statement(s);
}
```

```
while(condition) {
   while(condition)   {
     statement(s);
    }
  statement(s);
}
```

Nested loops (loop in a loop)

The syntax for a nested do...while loop statement is as follows −

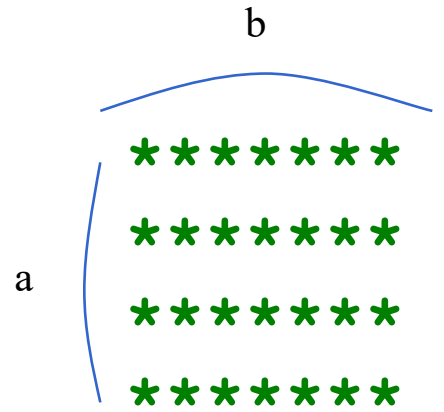
```
do {
   statement(s);
   do    {
     statement(s);
    } while(condition);
}while(condition);
```

Note- In loop nesting you can put any type of loop inside any other type of loop. For example, a 'for' loop can be inside a 'while' loop or vice versa.

Nested loops (loop in a loop)

In Nested loop one loop is place within another loop body.
When we need to repeated loop body itself n number of times use nested loops.
Nested loops can be design upto 255 blocks.

```
printf("%d%d",a,b); //a=4,b=7

for (i = 0; i < a; i++){
    for (j=0; j<b; j++) {
        printf("*");
    }
    printf("\n");
}
```

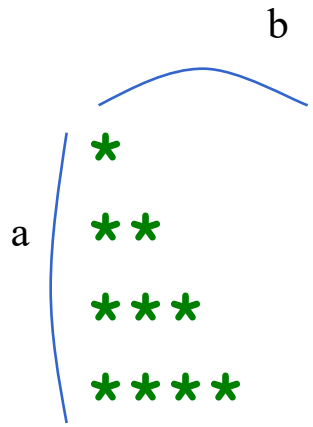


Nested loops (loop in a loop)

```
do{

   while(condition) {

      for ( initialization; condition; increment ) {

         // statement of inside for loop
      }

      // statement of inside while loop
   }

   // statement of outer do-while loop
}while(condition);
```

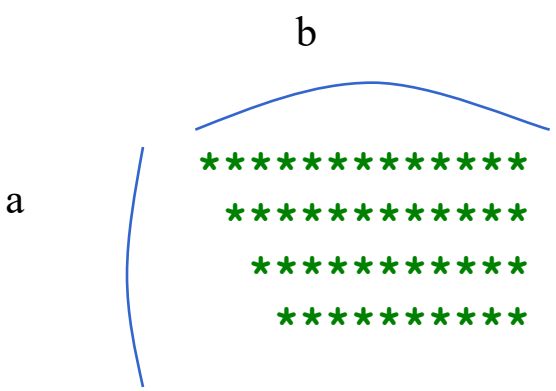
Nested loops example

```
int a,b;
Printf("Enter two values");
scanf("%d%d",&a,&b);

for (i = 0; i < a; i++) {
   for (j = 0; j < b; j++)
      if (j > i)
         break;
      printf("*");
   }
   printf("\n");
}
```



Nested loops example

```
int a,b;

scanf("%d%d",&a,&b);

for (int i = 0; i < a; i++) {
   for (int j=0; j<b; j++) {
      if (j < i)
         printf(" ");
      else
         printf("*");
   }
   printf("\n");
}
```



Nested loops example

```
int a,i,j;
scanf("%d",a);

for (i = 0; i < a; i++) {
   for (j=0; j<a; j++) {
      if (j < a-i)
         printf(" ");
      else  printf("*");
   }

   for (j=0; j<a; j++) {
      if (j > i)
         break;
      printf("*");
   }
printf("\n");
}
```