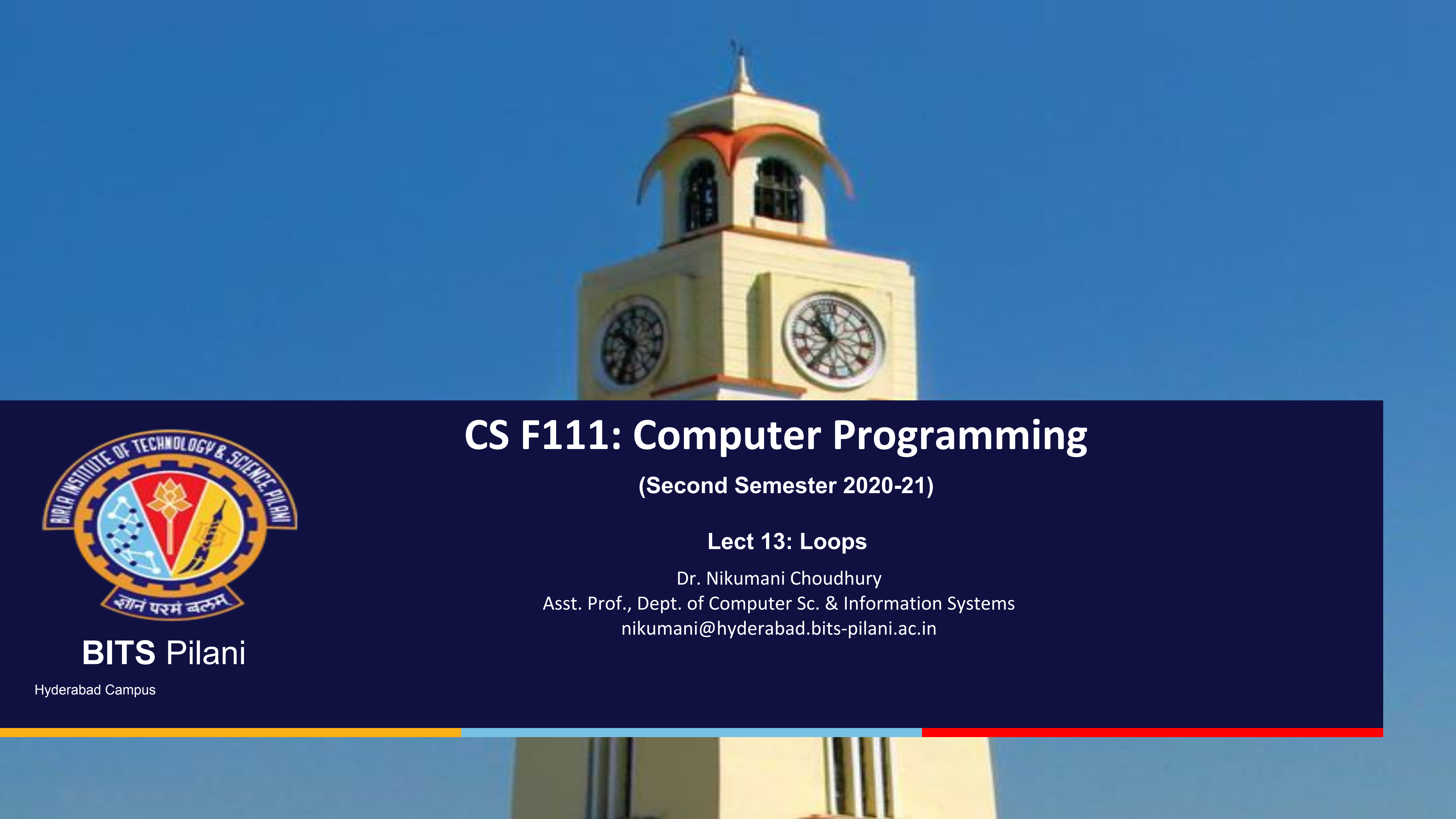


CS F111: Computer Programming

(Second Semester 2020-21)

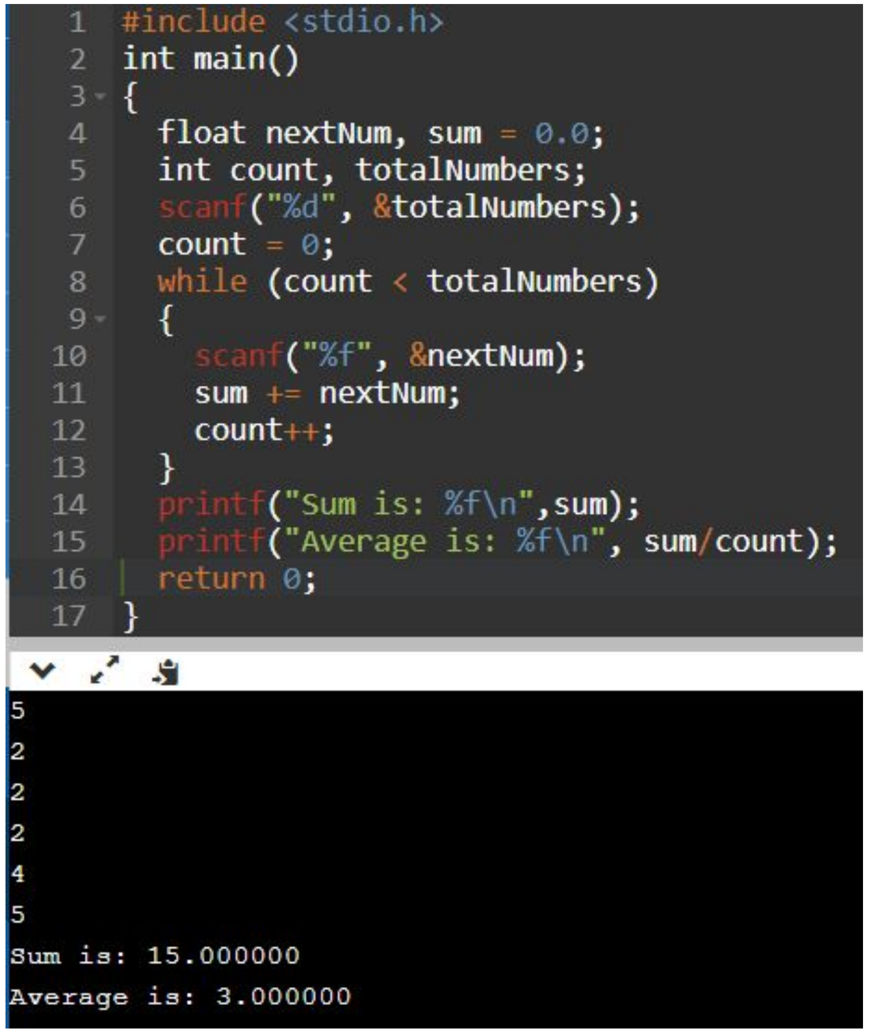
Lect 13: Loops

Dr. Nikumani Choudhury
Asst. Prof., Dept. of Computer Sc. & Information Systems
nikumani@hyderabad.bits-pilani.ac.in

BITS Pilani

Hyderabad Campus

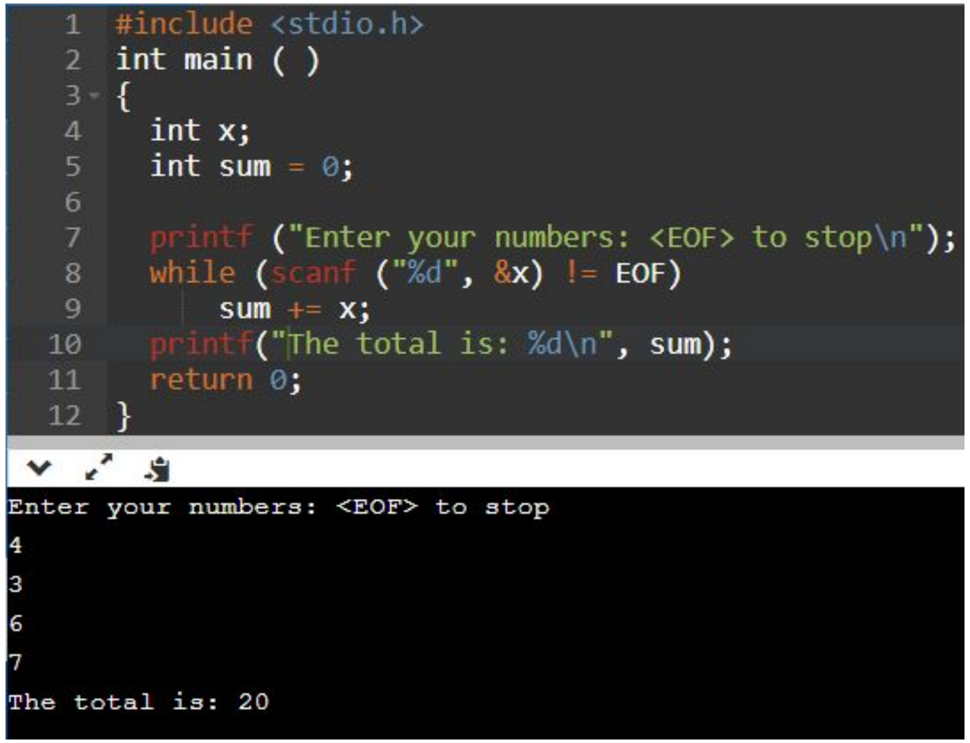
Examples: while and for



```c
#include <stdio.h>
int main()
{
   float nextNum, sum = 0.0;
   int count, totalNumbers;
   scanf("%d", &totalNumbers);
   count = 0;
   while (count < totalNumbers)
   {
      scanf("%f", &nextNum);
      sum += nextNum;
      count++;
   }
   printf("Sum is: %f\n",sum);
   printf("Average is: %f\n", sum/count);
   return 0;
}
```

```
5
2
2
2
4
5
Sum is: 15.000000
Average is: 3.000000
```

How will you do it with for?

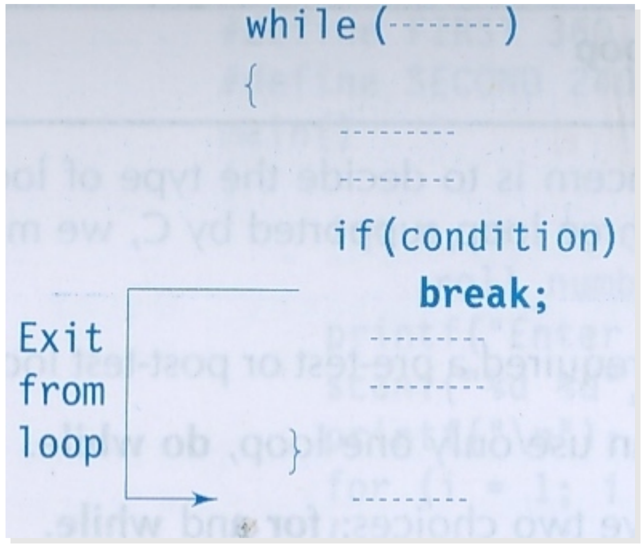


```c
#include <stdio.h>
int main ( )
{
   int x;
   int sum = 0;

   printf ("Enter your numbers: <EOF> to stop\n");
   while (scanf ("%d", &x) != EOF)
      sum += x;
   printf("The total is: %d\n", sum);
   return 0;
}
```

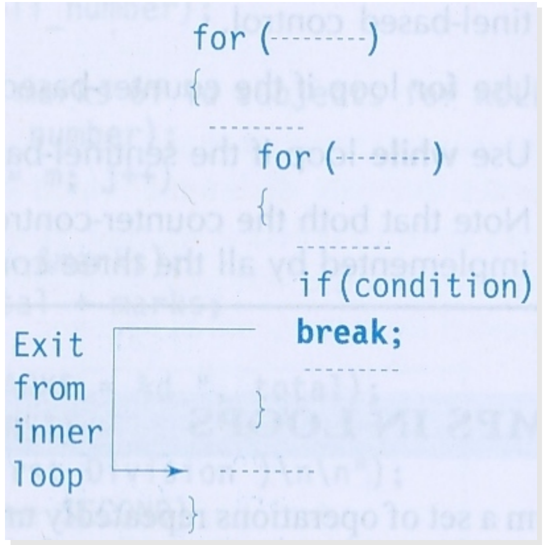
```
Enter your numbers: <EOF> to stop
4
3
6
7
The total is: 20
```

Jumps in Loops

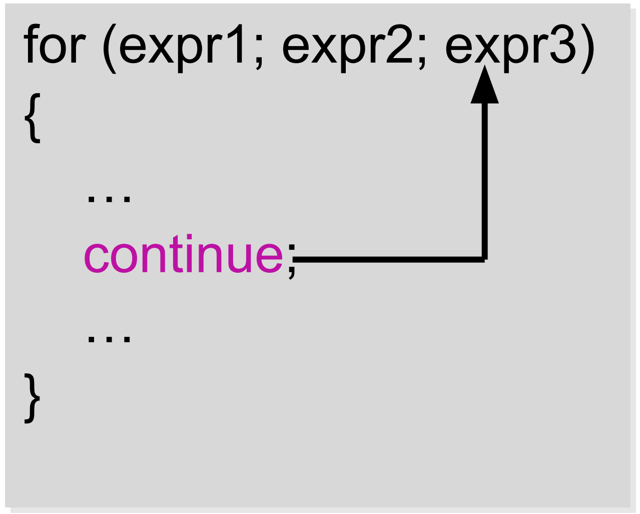
- May need to skip a part of loop or to leave the loop when a particular condition occurs
- For example, when a negative number is input, skip the rest all processing
- Jumping out of a loop: different cases



```
while (---------)
{
    ---------
    ---------
    if(condition)
        break;
    ---------
    ---------
}
---------
```



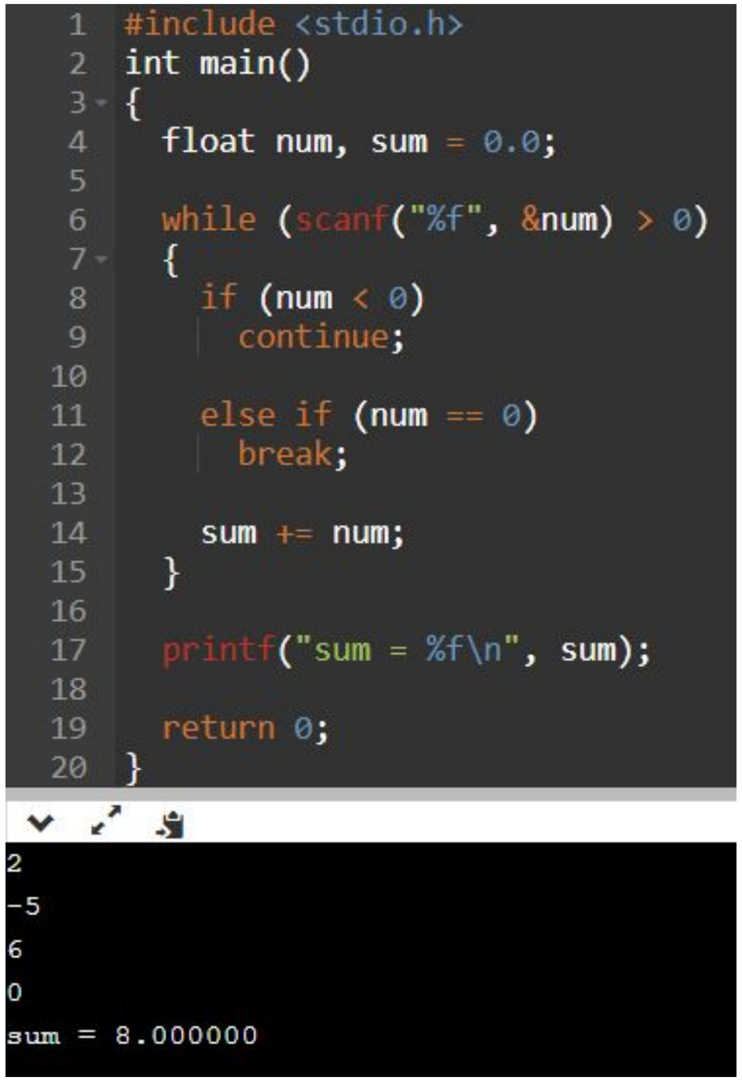
```
for (---------)
{
    ---------
    for (---------)
    {
        ---------
        if(condition)
        break;
        ---------
    }
    ---------
}
```



```
for (expr1; expr2; expr3)
{
    …
    continue;
    …
}
```

Examples

```
/* Average of nonzero nos*/
main()
{       int   count = 0;
        int   n;
        float avg;
        float sum = 0.0;
       while ( scanf ("%d", &n) != EOF)
       {
       if (n == 0)
           continue;
       sum += n;
       count++;
        }
        avg = sum / count;
        printf ("%f", avg);
}
```



```
#include <stdio.h>
int main()
{
  float num, sum = 0.0;

  while (scanf("%f", &num) > 0)
  {
    if (num < 0)
      continue;

    else if (num == 0)
      break;

    sum += num;
  }

  printf("sum = %f\n", sum);

  return 0;
}
```

```
2
-5
6
0
sum = 8.000000
```

Additional features of for Loop

- More than one variable can be initialized

```
p = 1;
for (n = 0; n < 5; ++n)
```

$\Updownarrow$

```
for (p = 1, n = 0; n < 5; ++n)
```

- Similarly, may have more parts in the increment section
- Test condition may have any compound relation and may not confine to loop control var

```
sum = 0;
for (i=1; i < 20 && sum < 100; ++i)
{
    sum = sum + i;
    printf("%d %d\n", i, sum);
}
```

- Initialization and increment sections may contain exprs

```
for ( x = (m+n) / 2; x > 0; x = x/2)
```

- One or more sections can be omitted

```
m = 5;
for (  ; m != 100 ;  )
{
    printf ("%d\n", m);
    m = m + 5;
 }
```

- Infinite loops are created by a null test condition

`for( ;  ; ) { …}` or `for( ; 1 ; ) {…}`

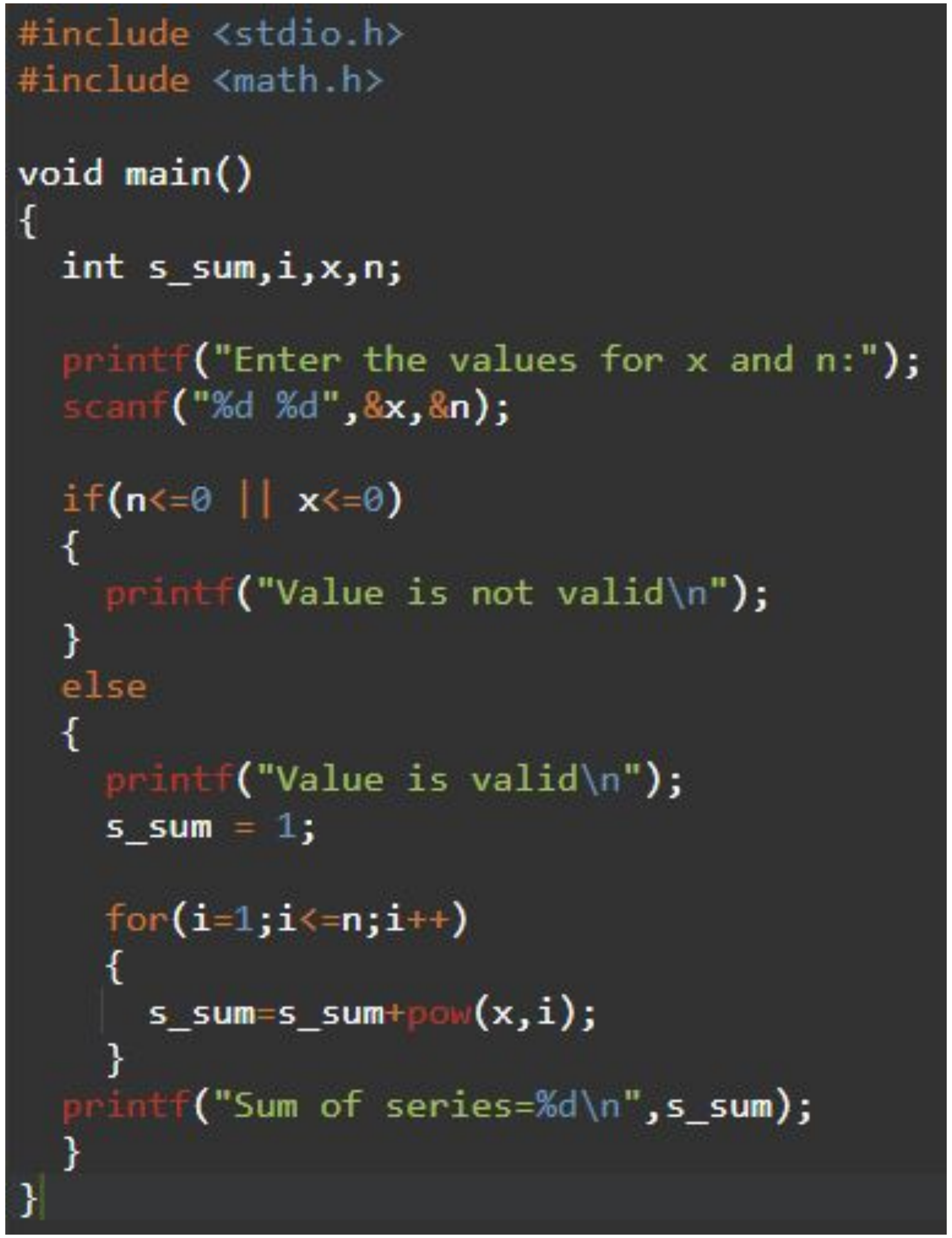
- Time delay loop is created by

```
for ( j = 1000; j > 0; j = j-1)
  ;
```

More Ex.

Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression:

$1+x+x^2+x^3+\ldots+x^n$



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

void main()
{
  int s_sum,i,x,n;

  printf("Enter the values for x and n:");
  scanf("%d %d",&x,&n);

  if(n<=0 || x<=0)
  {
    printf("Value is not valid\n");
  }
  else
  {
    printf("Value is valid\n");
    s_sum = 1;

    for(i=1;i<=n;i++)
    {
      s_sum=s_sum+pow(x,i);
    }
  printf("Sum of series=%d\n",s_sum);
  }
}
```

Examples continued…

Algorithm to print a series of numbers in the form of a right triangle.

```
1. Set line to 1
2. Loop (line not greater than limit)
    1. Set num to 1
    2. Loop (num !> line)
        1. Print num
        2. Increment num
    3. End loop
    4. Advance line
    5. Increment line
3. End loop
```

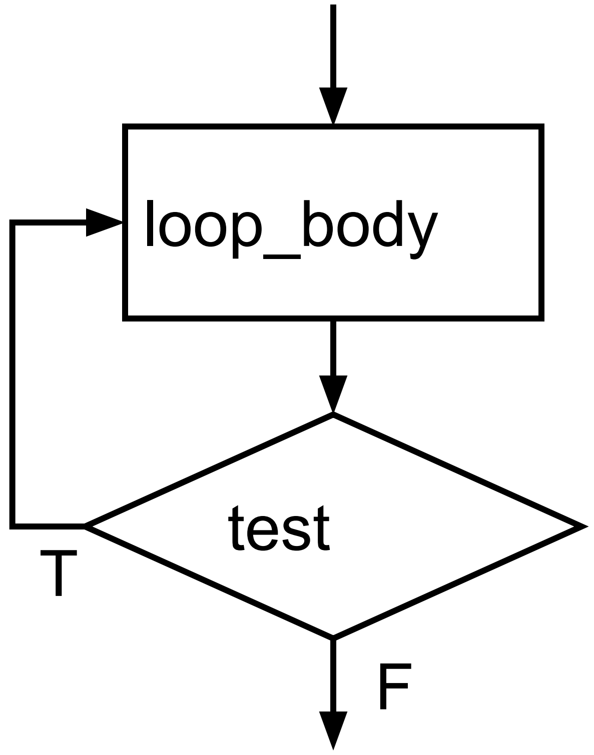
```
1
12
123
```

```
#include <stdio.h>
int main()
{
   int limit;
   printf ("Enter a no between 1 and 9:");
   scanf ("%d", &limit);
   for (int lineCtrl = 1; lineCtrl <= limit;
                                    lineCtrl++)
   {
      for (int numCtrl=1; numCtrl<=lineCtrl;
                                    numCtrl++)
           printf("%d", numCtrl);
      printf("\n");
   }
  return 0;
}
```

Nested loops

do…while

```
do
{
  loop_body;
}
while (test);
```

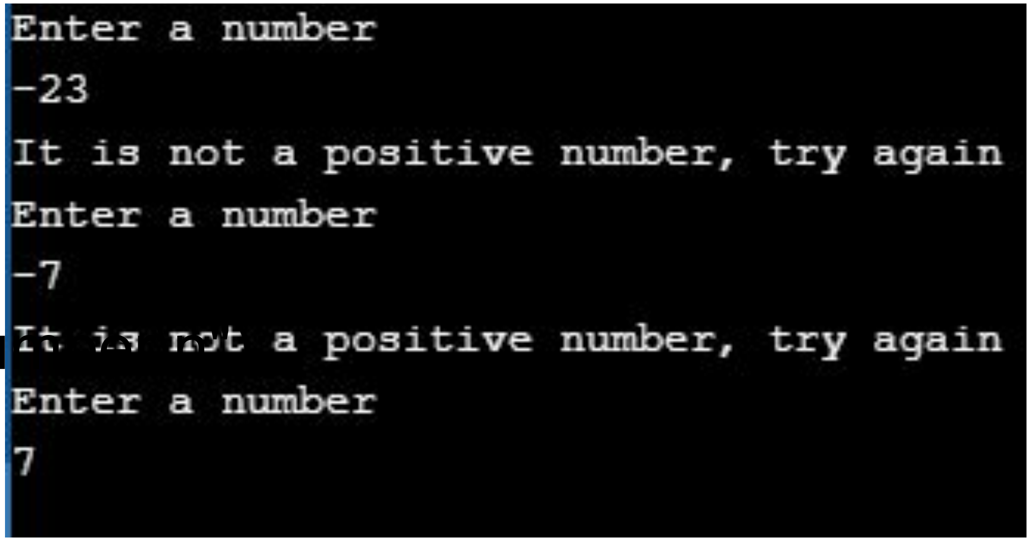


Executes loop body as long as test evaluates to TRUE (non-zero).

Note: Test is evaluated **after** executing loop body.

Example 1

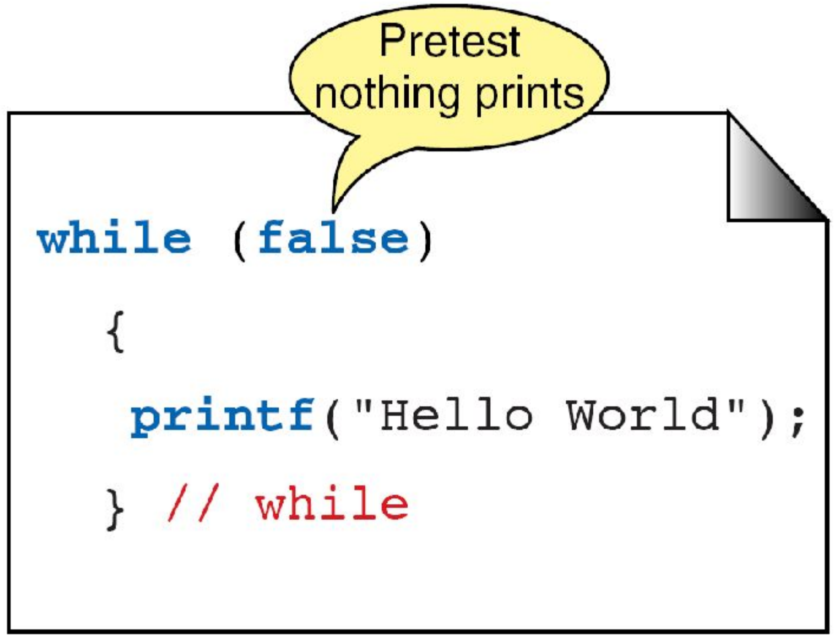
```
#include <stdio.h>
int main()
{
  int n;
  do
  {
        printf ("Enter a nu
        scanf("%d", &n);
        if (n <= 0)
           printf("It is not a positive number, try again\n");
   } while (n <= 0);

  return 0;
}
```



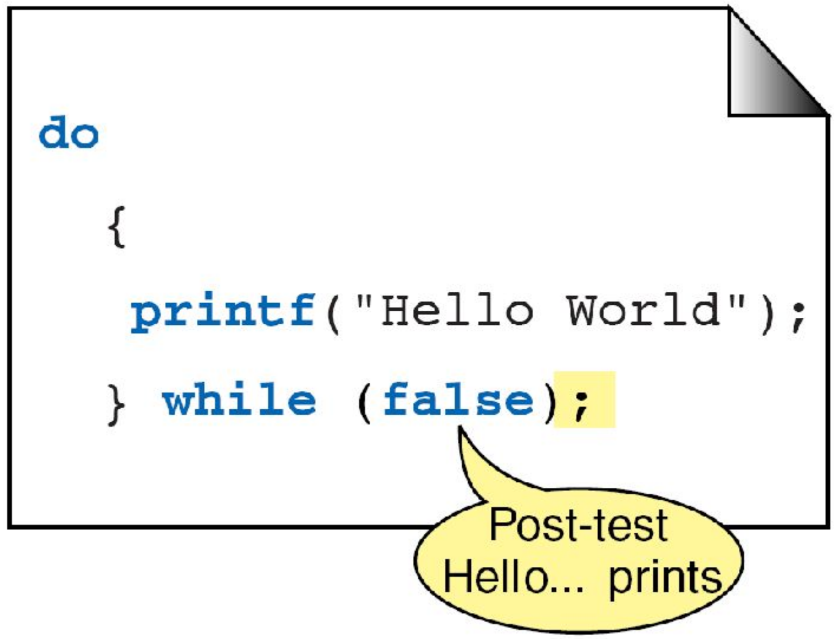
```
Enter a number
-23
It is not a positive number, try again
Enter a number
-7
It is not a positive number, try again
Enter a number
7
```

(Keep entering till you input a positive number)

while vs. do…while



```
while (false)
   {
    printf("Hello World");
   } // while
```



```
do
   {
    printf("Hello World");
   } while (false);
```

More Examples

```
#define COLMAX 10
#define ROWMAX 5

main()
 {
   int row,column, y;
   row = 1;
   printf("  MULTIPLICATION TABLE \n");
   do   /*......OUTER LOOP BEGINS........*/
   {
      column = 1;

      do   /*.......INNER LOOP BEGINS.......*/
      {
         y = row * column;
         printf("%4d", y);
         column = column + 1;
      }
      while (column <= COLMAX); /*... INNER LOOP ENDS ...*/

      printf("\n");
      row = row + 1;
   }
   while (row <= ROWMAX);/*.....   OUTER LOOP ENDS   .....*/
 }
```

```
MULTIPLICATION TABLE
1     2     3        …      10
2     4     6        …      20
3     6     9        …      30
4     8     12       …      40
5     10    15       …      50
```

```
int main()
{
   int x;
   int sum = 0;

   do
   {
      scanf("%d", &x);
      sum = sum + x;
    } while (x != 0);

    printf("%d", sum);
return 0;
}
```

Example with comma expr.

```c
#include <stdio.h>

int main (void)
{
// Local Declarations
   int loopCount;
   int testCount;

// Statements
   loopCount = 1;
   testCount = 0;
   printf("while loop:       ");
   while (testCount++, loopCount <= 10)
      printf("%3d", loopCount++);
   printf("Loop Count:       %3d\n", loopCount);
   printf("Number of tests:  %3d\n", testCount);

   loopCount = 1;
   testCount = 0;
   printf("\ndo...while loop:  ");
   do
      printf("%3d",    loopCount++);
   while (testCount++,  loopCount <= 10);

   printf("\nLoop Count:       %3d\n", loopCount);
   printf("Number of tests:  %3d\n", testCount);
   return 0;
}  // main
```

```
Results:
while loop:          1  2  3  4  5  6  7  8  9 10
Loop Count:         11
Number of tests:    11

do...while loop:     1  2  3  4  5  6  7  8  9 10
Loop Count:         11
Number of tests:    10
```

(Comma expression)

Better styles: avoid break

```
/* A bad loop style */
for (  ;    ;  )
{
   …
   if (condition)
     break;
}
```

```
/* A better loop style */
for (  ;  !condition ;    )
{
    …
}
```

```
/* A bad loop style */
while ( x )
{
   …
   if (condition)
         break;
   else
         …
}
```

```
/* A better loop style */
while ( x  &&  !condition)
{
    …
    if ( !condition )
      …;
}
```