The "for" Loop

Like "while" and "do/while" the "for" loop enables you to evaluate a sequence of expressions multiple numbers of times. For loops are best when you know the number of times that the expressions are needed to be evaluated in advance (counter-controlled repetition).

Syntax of “for” statement

```
for( initializations; condition; increment or decrement)
{
  Statements;
}
```

for: keyword

Initializations: You may initialize the counter and other variables here. This part is executed only once at the beginning.

Condition: Set the continuation-condition here to continue or end the loop. The condition is tested each time before the statements are executed.

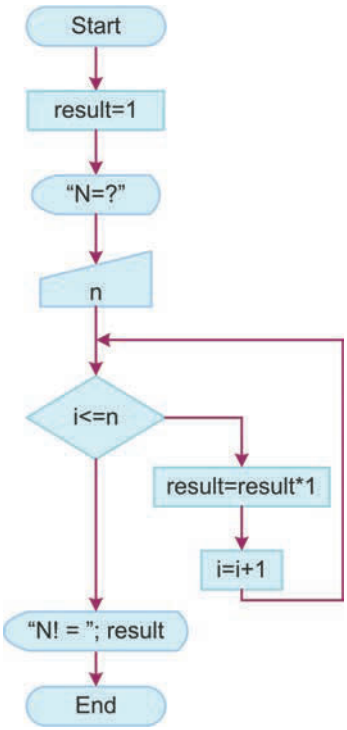
Increment or decrement: You may increment or decrement the counter here. This part is executed each time after the statements have been executed.

Statements: The statements are executed each time in the loop.

```
/*
PROG: c3_07for.cpp
Printing the numbers from 1 to 10 increasing by 1.
*/

#include <iostream>
using namespace std;
int main()
{
    int counter;
    for (counter=1; counter<=10; counter++)
    {
      cout<<counter<<" ";
    }

    system("PAUSE");
    return 0;
}
```

1 2 3 4 5 6 7 8 9 10 Press any key to continue . . .

Example: Factorial



Flowchart of the Program "factorial"

The **factorial** of a non-negative integer n is the product of the positive integers less than or equal to n. This is written as "n!", and pronounced "n factorial". 0!=1, 1!=1, 2!=2, 3!=6, 4!=24, 5!=120. The following program reads a non-negative integer from the standard input and computes its **factorial.**

```
/*
PROG: c3_08factorial.cpp
*/
#include<iostream>
using namespace std;

int main()
{
  int i, n;
  long long result=1;      //Initialize result for n=0 and n=1
                           //Factorial of numbers rapidly grow
                           //very large.
  cout<<"N=?";
  cin>>n;

  for(i=2; i<=n; i++)
    result *= i;           //result = result*i

  cout<<"N! = "<<result<<endl;

  system("pause");  return 0;
}
```

```
N=?11
N! = 39916800
Press any key to continue . . .
```

Example: X to the power of Y

The following program computes X raised to the power Y (X^Y) where X and Y are both integer numbers and Y is non-negative. X is the base (mantissa) and Y is the exponent value.

To get the result we have to multiply X, Y times. For example for `X=5` and `Y=3`, the result will be 5*5*5. In fact, C++ provides a built-in math "power" function (**double pow(double x, double y)**) that calculates power of integer or floating point numbers. We are going to use it in the chapter "Functions" in this book.

```
 /*
PROG: c3_09power.cpp
*/
#include <iostream>
using namespace std;

int main()
{
  int base, power;
  long long result = 1;    //Initial value for power = 0;

  cout <<"Enter the base and the power?";
  cin >> base >> power;    //Get the input

  //Calculate the result. Multiply base itself power times.
  for (int i=1; i<=power; i++)
    result *= base;

  //Print the result
  cout <<"Result is "<<result<<endl;

  system("pause");  return 0;
}
```

```
Enter the base and the power?4 3
Result is 64
Press any key to continue . . .
```

Example: Fibonacci Series

The Fibonacci series begins with 0 and 1 and has the property that each subsequent Fibonacci number is the sum of the previous two Fibonacci numbers. Some of the first Fibonacci numbers are 0, 1, 1, 2, 3, 5, 8, 13, …

Make a program to print the first N Fibonacci numbers where N is a positive integer.

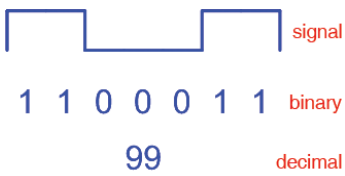
Leonardo Fibonacci (1170 - 1250) introduced to Europe and popularized the Hindu-Arabic number system (also called the decimal system). He contributed greatly to number theory.

```
/*
PROG: c3_10fibonacci.cpp
*/
#include <iostream>
using namespace std;

int main()
{
  int N;
  int currentNum=1;        //The first Fibonacci number
```

```
  int previousNum=0;

  cout <<"N =? ";
  cin >> N;

  cout<<previousNum<<" "; //Print the first Fibonacci number

  /*Set a loop to print the rest of the first N Fibonacci
  numbers*/
  for (int i=1; i<N; i++)
  {
    cout<<currentNum<<" ";       //Print the current Fibonacci
                                 //number
    currentNum += previousNum;   //CurrentNum gets the value
                                 //of the next Fib. number
    //previousNum gets the old value of currentNum.
    previousNum = currentNum - previousNum;
  }

  system("pause");
  return 0;
}
```

```
N =? 10
0 1 1 2 3 5 8 13 21 34 Press any key to continue . . .
```

Example: Binary to Decimal



Binary is a number system used by digital devices like computers. The computer represents values using two voltage levels (usually 0V and +5V). With two levels we can represent exactly two different values. These could be any two different values, but by convention we use the values zero (0) and one (1).

To convert binary into decimal is very simple. Just like the decimal system, we multiply each digit by its weighted position, and add each of the weighted values together. For example, the binary value 10011101 represents:

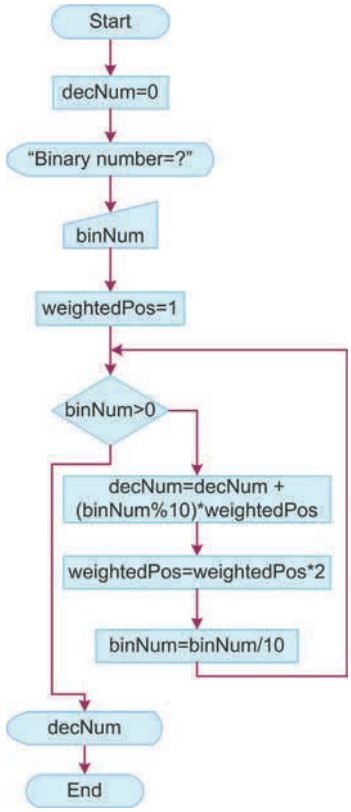
$1*2^7 + 0*2^6 + 0*2^5 + 1*2^4 + 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 157$.

Make a program that gets a binary number and prints its decimal equivalent as output.

```
/*
PROG: c3_11bintodec.cpp
*/
#include <iostream>
using namespace std;
```

```
int main()
{
  long long binNumber, decNumber = 0;

  cout<<"Enter the binary number:";
  cin >> binNumber;

  int weightedPos = 1;     //The weighted position of the
                           //right-most digit
  while (binNumber > 0)    //We have some more digits to
                           //process
  {
    /*Multiply the right-most digit by its weighted position
    and add the product to decNumber.*/
    decNumber += (binNumber%10)*weightedPos;
    weightedPos *= 2;      //The weighted position of the next
                           //digit
    binNumber /= 10;       //Cut out the right-most digit of
                           //binNumber.
  }

  cout<<"Decimal equivalent is "<<decNumber<<endl;
  system("pause");
  return 0;
}
```

```
Enter the binary number:1110011
Decimal equivalent is 115
Press any key to continue . . .
```



Flowchart of the Program "bintodec"