for loop

- The initialization is executed once, before the loop is entered.
- The condition is checked before each iteration
- The update is executed after each iteration.

syntax

```
for (initialization; condition; update)
{
  statements ;
}
```

1 2 4 3

```
int counter;
int fact = 1;
for ( counter = 1; counter <= 3; counter++)
{
      fact = fact*counter ;
}
```

1 <= 3 → True.
2 <= 3 → True.
3 <= 3 → True
4 <= 3 → False.

fact	counter
1	1.
2	2.
6	3
	4.

fact = 1 * 1 = 1
fact = 1 * 2 = 2.
fact = 2 * 3 = 6

What is the output of following program (when embedded in a complete program?)

```
for (int count= 1; count < 5; count++)
{
    cout<<(2*count) << " ";
}
```

count
1
2
3
4
5

1 < 5 → True.
2 < 5 → True.
3 < 5 → True
4 < 5 → True
5 < 5 → False

OUTPUT

```
2 4 6 8
```

What is the output of following program (when embedded in a complete program?)

```
for (int n = 6; n > 0; n=n-2)
{
      cout<<" Hello " ;
      cout<<n<<endl ;
}
```

n
6
4.
2.
0.

6 > 0 → True
4 > 0 → True.
2 > 0 → True
0 > 0 → False

OUTPUT.

```
Hello 6
Hello 4
Hello 2
```

```
for (int i=1 ; i<=5 ; i++)
{
    cout << i << endL ;
}
```

⇒

OUTPUT

```
1
2
3
4
5
```

```
for (int i=1 ; i<5 ; i++)
{
    cout << i << endL ;
}
```

OUTPUT

```
1
2
3
4
```

inputs → 20, 30, 40

20 + 30 + 40 = 90 (sum)

$\frac{90}{3} = 30$ (Average)

A Program to calculate the Sum and Average from ~~5~~ 3 integer numbers entered by User.

input	sum	counter
20	0	0
30	20	1
40	50	2
	90	3

```
#include <iostream>
using namespace std;

int main()
{
    int counter = 0, input, sum = 0, average;
    cout<<"Enter sequence of integer values:";
    while (cin>>input)
    {
        sum = sum + input;
        counter++;
    }

    average = sum / counter;
    cout<<"Sum: "<<sum<<" Average: "<<average;
}
```

sum = 0 + 20 = 20

sum = 20 + 30 = 50

sum = 50 + 40 = 90

$= \frac{90}{3} = 30$.

(sum → 90, average → 30)

OUTPUT

```
Enter sequence of integer numbers: 20 30 40.
Sum: 90 Average: 30.
```

largest = ~~23~~ ~~78~~ (90)

23, 78, 90, 56

78 90 90

Finding Largest Number

- To compute the largest value in a sequence, keep a variable that stores the largest element that you have encountered, and update it when you find a larger one.

A program to find largest number from 4 integer values entered by the user.

23 78 90 56

```
#include <iostream>
using namespace std;

int main()
{
    cout<<"Enter sequence of integer values: ";
    int largest;
    cin>>largest;
    int input;
    while (cin>>input)
    {
        if (input > largest)
        {
            largest = input;
        }
    }

    cout<<"The largest value is:"<<largest;
}
```

Largest	input
~~23~~ ~~78~~ 90	78
	90
	56

23 > 78 → False.

78 > 23 → True.

90 > 78 → True.

56 > 90 → False.

Enter sequence of integer values: 23 78 90 56.
The largest value is: 90.

10.111

Binary point.

Review on Numbering System

- The binary number system has base 2.
- The value of digit is determined by its position in the number.
- The two binary digits are: 1 and 0.

Weight structure:

Binary point.

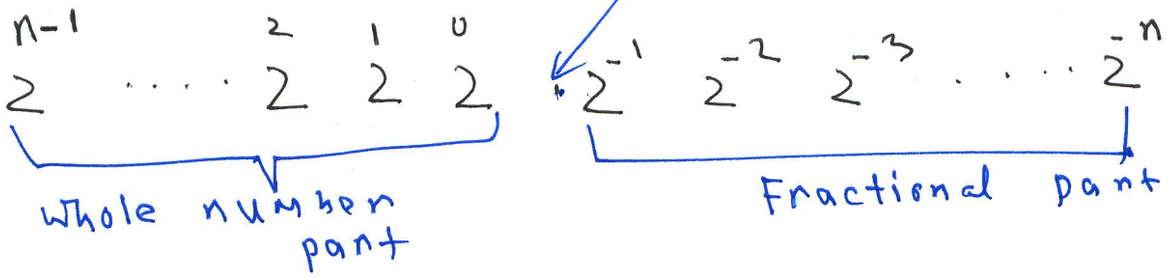


TABLE 2-2

Binary weights.

Positive Powers of Two (Whole Numbers)									Negative Powers of Two (Fractional Number)					
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
256	128	64	32	16	8	4	2	1	1/2 0.5	1/4 0.25	1/8 0.125	1/16 0.625	1/32 0.03125	1/64 0.015625

Example 2: Convert 10.111 to decimal number.

Binary Number: 1 0 . 1 1 1

Weights : 2^1 2^0 2^{-1} 2^{-2} 2^{-3}

$= (1)\,2^1 + 2^{-1} + 2^{-2} + 2^{-3}$

$= 2 + 0.5 + 0.25 + 0.125$

$= 2.875$

Converting whole decimal numbers to binary (Sum-of-weights Method)

- Determine the set of binary weights whose sum is equal to the decimal number.
- Place 1's and 0's on the appropriate weight positions determines the binary number for that decimal number.

Example 1: Convert decimal number 25 to binary using Sum-of-Weights Method

	4	3	2	1	0
Weight)	2	2	2	2	2
weight)	16	8	4	2	1
	1	1	0	0	1

16 + 8 + 1 = 25

$25_{10} = 11001_2$

Example 2: Convert decimal number 58 to binary using Sum-of-Weights Method.

Hexadecimal Numbers

- The hexadecimal number system has sixteen characters; it is used primarily as a compact way of displaying or writing binary numbers because it is very easy to convert between binary and hexadecimal.

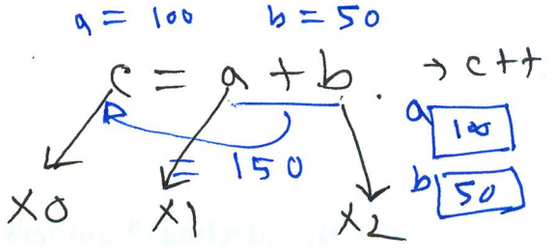
TABLE 2-3

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Convert the following binary numbers to hexadecimal: (a) 1100101001010111

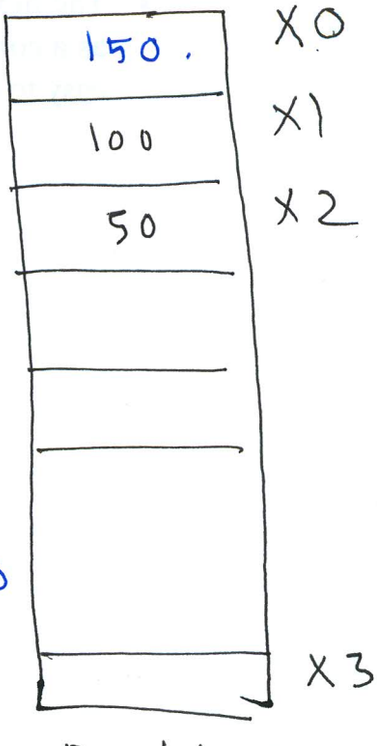
Determine the binary numbers for the following hexadecimal numbers: (a) $10A4_{16}$

Chapter 2



ADD

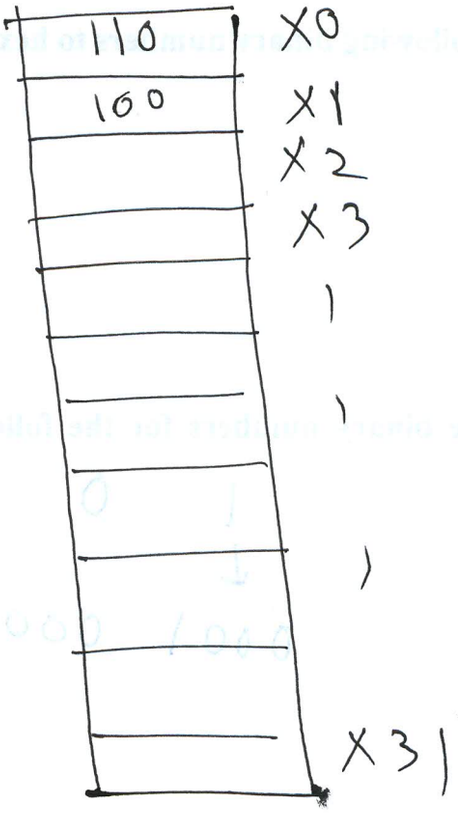
ADD X0, X1, X2 //c = a + b



Register.

ADDI

ADD X0, X1, #10 //X0 = X1 + 10



→ 64 bit