

Answer 1 : Calculate to sum of the numbers between 0 and entered number.

Source Code :

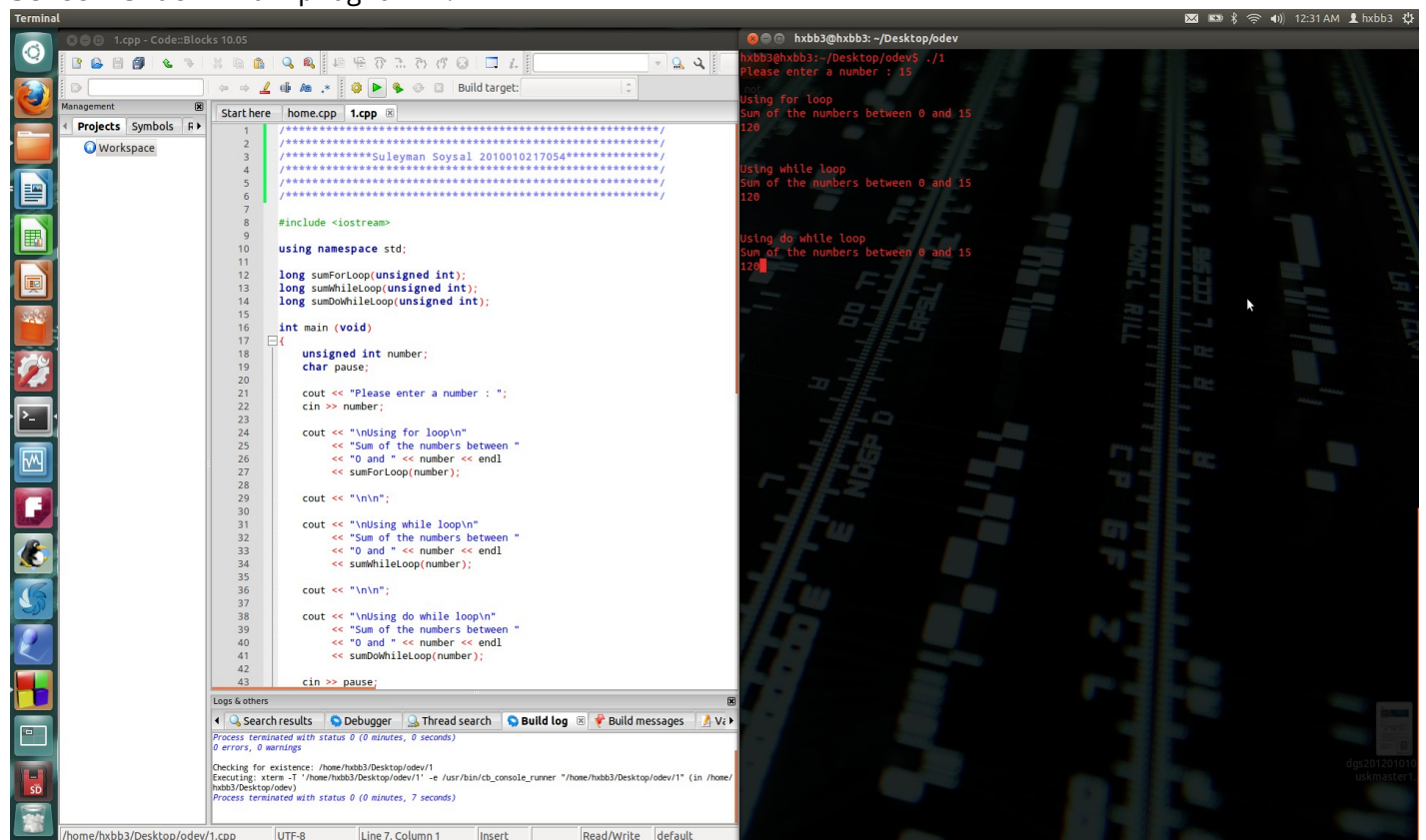
```
1  /*****  
2  /*****  
3  /*****titaNWare*****/  
4  /*****x_titanyum_x@hotmail.com*****/  
5  /*****  
6  /*****  
7  
8  #include <iostream>  
9  
10 using namespace std;  
11  
12 long sumForLoop(unsigned int);  
13 long sumWhileLoop(unsigned int);  
14 long sumDowhileLoop(unsigned int);  
15  
16 int main (void)  
17 {  
18     unsigned int number;  
19     char pause;  
20  
21     cout << "Please enter a number : ";  
22     cin >> number;  
23  
24     cout << "for loop"  
25         << "Sum of the numbers between "  
26         << "0 and " << number << endl  
27         << sumForLoop(number);  
28  
29     cout << "  
30  
31     cout << "while loop"  
32         << "Sum of the numbers between "  
33         << "0 and " << number << endl  
34         << sumWhileLoop(number);  
35  
36     cout << "  
37  
38     cout << "do while loop"  
39         << "Sum of the numbers between "  
40         << "0 and " << number << endl  
41         << sumDowhileLoop(number);  
42  
43     cin >> pause;  
44     return 0;  
45 }  
46  
47 long sumForLoop(unsigned int number)  
48 {  
49     unsigned int counter;  
50     long sum=0;  
51  
52     for(counter=number;counter>0;counter--)  
53     {  
54         sum += counter;  
55     }  
56     return sum;  
57 }  
58  
59 long sumWhileLoop(unsigned int number)  
60 {  
61     unsigned int counter;  
62     long sum=0;  
63  
64     counter=number;  
65     while(counter > 0)  
66     {  
67         sum += counter;  
68         counter--;  
69     }  
70     return sum;  
71 }
```

```

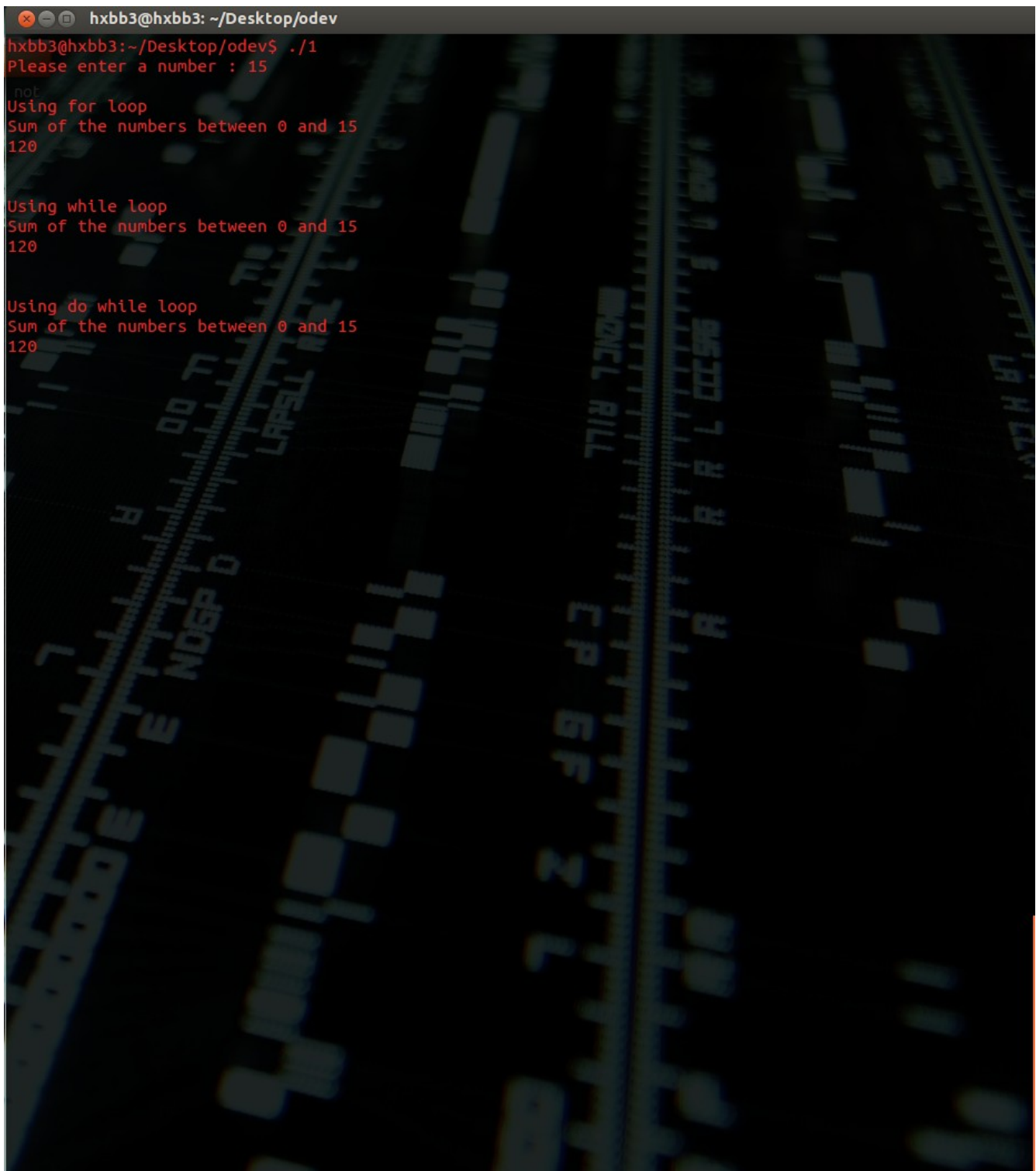
72
73 long sumDowhileLoop(unsigned int number)
74 {
75     unsigned int counter;
76     long sum=0;
77
78     counter = number;
79     do
80     {
81         sum += counter;
82         counter--;
83     }while(counter > 0);
84
85     return sum;
86 }
87
88
89
90

```

Screen shot 1 for program 1:



Screen shot for program 1 :



```
hxbb3@hxbb3: ~/Desktop/odev
hxbb3@hxbb3:~/Desktop/odev$ ./1
Please enter a number : 15
not
Using for loop
Sum of the numbers between 0 and 15
120

Using while loop
Sum of the numbers between 0 and 15
120

Using do while loop
Sum of the numbers between 0 and 15
120
```

The image shows a terminal window with a dark background and light-colored text. The window title is 'hxbb3@hxbb3: ~/Desktop/odev'. The user has entered the command './1' and the program has prompted 'Please enter a number : 15'. The program then outputs 'not', followed by three sections of code execution: 'Using for loop', 'Using while loop', and 'Using do while loop'. Each section outputs 'Sum of the numbers between 0 and 15' and the result '120'.

Answer 2 : Calculate $x \& y$, $x \wedge y$, $x | y$, $\sim x$ for $x=3AB$ and $y=FC9$.

Source code :

```
1  /*****titanWare*****/
2  /*****x_titanyum_x@hotmail.com*****/
3  /*****x_titanyum_x@hotmail.com*****/
4  /*****x_titanyum_x@hotmail.com*****/
5  /*****x_titanyum_x@hotmail.com*****/
6  /*****x_titanyum_x@hotmail.com*****/
7
8  #include <iostream>
9  #include <cstdio>
10
11 using namespace std;
12
13 void printScreenBinary(unsigned int);
14
15 int main (void)
16 {
17     unsigned int hexNumber1 = 0x3AB;
18     unsigned int hexNumber2 = 0xFC9;
19     unsigned int andOp;
20     unsigned int orOp;
21     unsigned int xorOp;
22     unsigned int complementOp;
23     unsigned int pause;
24
25     andOp      = hexNumber1 & hexNumber2;
26     orOp       = hexNumber1 | hexNumber2;
27     xorOp      = hexNumber1 ^ hexNumber2;
28     complementOp = ~hexNumber1;
29
30     printf("Number1... = %X= %d", hexNumber1, hexNumber1);
31     cout << "Binary = ";
32     printScreenBinary(hexNumber1);
33
34     cout << endl << endl;
35
36     printf("Number2... = %X= %d", hexNumber2, hexNumber2);
37     cout << "Binary = ";
38     printScreenBinary(hexNumber2);
39
40     cout << endl << endl;
41     cout << "-----";
42     cout << "Number1 & Number2 ";
43     cout << "Number 1 = " ;printScreenBinary(hexNumber1);
44     cout << "Number 2 = " ;printScreenBinary(hexNumber2);
45     cout << "Result is = ";printScreenBinary(andOp);
46     cout << "-----";
47
48     cout << endl << endl;
49     cout << "-----";
50     cout << "Number1 | Number2 ";
51     cout << "Number 1 = " ;printScreenBinary(hexNumber1);
52     cout << "Number 2 = " ;printScreenBinary(hexNumber2);
53     cout << "Result is = ";printScreenBinary(orOp);
54     cout << "-----";
55
56     cout << endl << endl;
57     cout << "-----";
58     cout << "Number1 ^ Number2 ";
59     cout << "Number 1 = " ;printScreenBinary(hexNumber1);
60     cout << "Number 2 = " ;printScreenBinary(hexNumber2);
61     cout << "Result is = ";printScreenBinary(xorOp);
62     cout << "-----";
63
64     cout << endl << endl;
65     cout << "-----";
66     cout << "~Number1";
67     cout << "Number 1 = " ;printScreenBinary(hexNumber1);
68     cout << "Result is = ";printScreenBinary(complementOp);
69     cout << "-----";
70
```

```

71     cin >> pause;
72     return 0;
73 }
74
75 void printScreenBinary(unsigned int number)
76 {
77     unsigned int counter;
78     unsigned int mask = 1 << 31;
79
80     for(counter=1;counter<=32;counter++)
81     {
82         if(number & mask) cout << "1";
83         else cout << "0";
84         number <<=1;
85
86         if(counter % 8 == 0)
87             cout << " ";
88     }
89     cout << "\n";
90 }
91
92

```

Screen shot for program 2 :

The screenshot shows a C++ IDE with the following code in `home.cpp`:

```

1 //*****Suleyman Soysal 2010010217054*****
2
3 //*****Suleyman Soysal 2010010217054*****
4
5 //*****Suleyman Soysal 2010010217054*****
6
7
8 #include <iostream>
9 #include <stdio>
10
11 using namespace std;
12
13 void printScreenBinary(unsigned int);
14
15 int main (void)
16 {
17     unsigned int hexNumber1 = 0x3AB;
18     unsigned int hexNumber2 = 0xFC9;
19
20     unsigned int orOp;
21     unsigned int xorOp;
22     unsigned int complementOp;
23     unsigned int pause;
24
25     andOp = hexNumber1 & hexNumber2;
26     orOp = hexNumber1 | hexNumber2;
27     xorOp = hexNumber1 ^ hexNumber2;
28     complementOp = ~hexNumber1;
29
30     printf("Number1... \nHexadecimal = %X\nInteger = %d\n",hexNumber1,hexNumber1);
31     cout << "Binary = ";
32     printScreenBinary(hexNumber1);
33
34     cout << endl << endl;
35
36     printf("Number2... \nHexadecimal = %X\nInteger = %d\n",hexNumber2,hexNumber2);
37     cout << "Binary = ";
38     printScreenBinary(hexNumber2);
39
40     cout << endl << endl;
41     cout << "-----\n";
42     cout << "Number1 & Number2 \n";
43     cout << "Number 1 = " << printScreenBinary(hexNumber1);

```

The terminal output shows the execution results:

```

Number1...
Hexadecimal = 3AB
Integer = 939
Binary = 00000000 00000000 00000011 10101011

Number2...
Hexadecimal = FC9
Integer = 4041
Binary = 00000000 00000000 00001111 11001001

-----
Number1 & Number2
Number 1 = 00000000 00000000 00000011 10101011
Number 2 = 00000000 00000000 00001111 11001001
Result is = 00000000 00000000 00000011 10001001

-----
Number1 | Number2
Number 1 = 00000000 00000000 00000011 10101011
Number 2 = 00000000 00000000 00001111 11001001
Result is = 00000000 00000000 00001111 11101011

-----
Number1 ^ Number2
Number 1 = 00000000 00000000 00000011 10101011
Number 2 = 00000000 00000000 00001111 11001001
Result is = 00000000 00000000 00001100 01100010

-----
~Number1
Number 1 = 00000000 00000000 00000011 10101011
Result is = 11111111 11111111 11111100 01010100

```


Screen shot 2 for program 2 :

```
hxbb3@hxbb3: ~/Desktop/odev
hxbb3@hxbb3:~/Desktop/odev$ ./home
Number1...
Hexadecimal = 3AB
Integer = 939
Binary = 00000000 00000000 00000011 10101011

Number2...
Hexadecimal = FC9
Integer = 4041
Binary = 00000000 00000000 00001111 11001001

-----
Number1 & Number2
Number 1 = 00000000 00000000 00000011 10101011
Number 2 = 00000000 00000000 00001111 11001001
Result is = 00000000 00000000 00000011 10001001
-----

Number1 | Number2
Number 1 = 00000000 00000000 00000011 10101011
Number 2 = 00000000 00000000 00001111 11001001
Result is = 00000000 00000000 00001111 11101011
-----

Number1 ^ Number2
Number 1 = 00000000 00000000 00000011 10101011
Number 2 = 00000000 00000000 00001111 11001001
Result is = 00000000 00000000 00001100 01100010
-----

~Number1
Number 1 = 00000000 00000000 00000011 10101011
Result is = 11111111 11111111 11111100 01010100
-----
```