NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

SCHOOL OF MECHANICAL AND MANUFACTURING ENGINEERING

CS-114 - Fundamental of Programing

LAB MANUAL # 4

ME -15 (C)

SUBMITTED BY:

MOHAMMAD GULZAIB

SUBMITTED TO:

SIR M. AFFAN

REPETITION STRUCTURE AND LOOPS

Repetition structure or loop is used to repeat the same process again and again until the required conditions are fulfilled. The programmer sets the conditions for the loop and the computer runs the loop under specific conditions. The loop will continue to run until the answer of condition comes out to be yes. Once the condition's answer comes out as false, the loop will terminate automatically, and loop ends.

Types of Repetition Structure:

There are two types of repetition structures:

1. Pretest
2. Post-Test loops

Pretest:

In this type of loops the condition is written at the beginning of loop. The loop first check that condition and then goes to loop. If loop is true, then main body will run and every time the loop run there will be change in the variable. If we did not make any change in value of variable then the loop will become infinite loop. For example, while loop, for loop, etc.

Post Test loop:

The main and only difference between these two types of loops is that posttest loop first runs and then checks the condition. It means that the loop body will run at least one time irrespective of the condition is true or not. For example, do..while loop.

For loop:

This loop will has defined starting point, defined condition, and defined final point. In for loop, it also has the increment in the variable, which is used to check the condition.

Syntax:

```
for (starting point; condition; increment)
  loop body
```

While loop:

While loop has only condition. It does not require increment to run. However, in order to get the required result, we need to define the increment, otherwise it will continue to loop.

Syntax:

```
while (condition)
    loop body
```

1. **Write a program in C++ that prints the numbers from 1 to 150 except the multiples of 10. Make use of the continue statement.**

   For this program we can use both loops while loops and for loop. Also in for loop we can write this program in two ways wither by using continue statement or by using if statement.

   Code:

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

int main(){
    int x;
    cout<<"The numbers are "<<endl;
    for(x=1;x<=150;x++){    //for loops starts from 1 and x will remain less than or equal to 150, everytime x will be added by 1
    if(x%10==0){            //when x equals multiples of 10, if statement runs and will take it back to loop without output
        continue;
    }
    cout<<x<<endl;        //outputs every number except multiples of 10
}
    return 0;
}
```

Without continue statement:

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

int main(){
    int x;
    for(x=1;x<=150;x++){ //it will start from 1 and continue till 150


    if(x%10== 0){  //only divisible of 10 will make this if statement to run
        x=x+1;     //if x is equal to any divisible of 10, 1 will be added to it
        if(x>=150){ //if x equals 150 this will run and it will break the loop and ends thw program
            break;
        }
    }
        cout<<x<<endl;  //shows all numbers except multiples of 10 till 150
}
    return 0;
}
```

Result:

```
The numbers are
1
2
3
4
5
6
7
8
9
11
12
13
14
15
16
17
18
19
21
22
23
24
25
26
27
28
29
31
32
```

```
31
32
33
34
35
36
37
38
39
41
42
43
44
45
46
47
48
49
51
52
53
54
55
56
57
58
59
61
62
63
```

```
62
63
64
65
66
67
68
69
71
72
73
74
75
76
77
78
79
81
82
83
84
85
86
87
88
89
91
92
93
94
```

```
93
94
95
96
97
98
99
101
102
103
104
105
106
107
108
109
111
112
113
114
115
116
117
118
119
121
122
123
124
125
```

```
124
125
126
127
128
129
131
132
133
134
135
136
137
138
139
141
142
143
144
145
146
147
148
149

----------
```

2. **Write a C++ program to find the sum of digits of a number.**

This program will take each digit of entered number and add all the digit and give the result. For statement will be used, it will divide the provided number by 10 until we get zero and then the loop will break. The remainder we get when divided by 10 will be added into each other and we get the sum of all digits.

Code:

```
#include<iostream>
using namespace std;
int main(){
    int x;                          //variable to which the number whose sum of is required will be assigned
    cout<<"Enter the number ";
    cin>>x;
    int s=0;                         //'s' is declared equal to zero, which will be used in for loop to store value of sum of digits
    int y,z;
    z=x;
for(z;z>0;z=z/10){                  //x will be given by user and this loop will run until x comes out to be zero, and evertime loop runs x will be divided
    y=z%10;                         //the first remainder will be first digit from left, stored in y and then added in s, loop runs and add digits
    s=s+y;
    cout<<y;
    if(z>10)
    cout<<"+";
}
    cout<<" = "<<s;                      //output
return 0;
}
```

Result:

```
Enter the number 5678
8+7+6+5 = 26
--------------------------------
Process exited after 4.036 seconds with return value 0
Press any key to continue . . .
```

3. **Write a program in C++ to check whether a number is prime or not.**

This function will be solved using for loop. As we know that prime numbers only divisible by itself and 1. So we will give condition such that the given number will be divided by every number until the number which is entered is arrived. If the number is divided by any number, then a third variable whose initial value is zero, will be added by one. Each time a number gets divided by any number 1 will be added to the third variable. It means if a number is only divisible by 1 and itself then we will get 2. Only that number will be prime number and if we get two then output will be prime number and not prime number if any number other than 2 will be assigned to third variable.

Code:

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

int main(){
    int z,b=0;
    cout<<"Enter the number you want to get information about ";
    cin>>z;
    int y=1;      //loop variable declared outside
    for(y;y<=z;y++){  //loop will check until y equals the input number

    if(z%y==0){ //every time z%y equals zero then 1 will be added to b
        b++;
    }
    }
    if(b==2)  //if the number is prime then b=2, that is prime is only divided by two numbers
    {
        cout<<z<<" is a prime number";  //if b=2 then cout prime
    }
    else
    cout<<z<<" is not a prime number"; //if b is not equal 2 that is z is not a prime nmbr
    return 0;
}
```

Result:

```
Enter the number you want to get information about 131
131 is a prime number
--------------------------------
Process exited after 8.349 seconds with return value 0
Press any key to continue . . .
```

```
Enter the number you want to get information about 566
566 is not a prime number
--------------------------------
Process exited after 4.331 seconds with return value 0
Press any key to continue . . .
```