Contents

I Loop Control Using For Loop **2**

1 Print All ASCII Values . . . 3
2 Print first n natural numbers . . . 3
3 Multiplication table of a number . . . 3
4 Leap Years in a Range . . . 3
5 Power of a number . . . 3
6 Count number of digits in a number . . . 4
7 Calculate the sum of digits of a number and reverse it. . . . 4
8 Evaluate the sum of the series: 1-x^2+x^4-x^6.... upto N terms . . . 4
9 Prime Number Checking . . . 5

II Loop Control- Use while Loop **6**

10 Check Whether Given Number is Palindrome or Not . . . 7
11 Factorial of a Number . . . 7
12 Factors of a number . . . 7
13 Check whether a number is Armstrong Number . . . 7
14 Check whether a number is Perfect Number . . . 7
15 Check whether a number is Strong Number . . . 8
16 Generate Fibanocci Series . . . 8
17 Prime factors of a number . . . 8
18 Display Floyd's triangle . . . 8

III Loop Control- Use do while loop and switch case **10**

19 Menu Driven Program to implement simple calculator . . . 11
20 Multiplication table . . . 11
21 Menu driven program to display :
Armstrong Numbers, Prime Series, Fibonacci Series, Perfect Numbers and Strong Numbers in a range or limit . . . 11
22 Menu driven program to display the following patterns:Even Number Pyramid , Pyramid of Multiplication Tables . . . 11

Part I

Loop Control Using For Loop

1 Print All ASCII Values

Print characters corresponding to the integers from 1 to 256. Use for loop.

2 Print first n natural numbers

Read the number of natural numbers to be printed into variable n
Print the natural numbers from 1 to n using for loop

3 Multiplication table of a number

Read the number upto which multiplication table has to be printed.
Sample Output

```
Enter an integer to find multiplication table: 6
Enter range of multiplication table: 4
6 * 1 = 6
6 * 2 = 12
6 * 3 = 18
6 * 4 = 24
```

4 Leap Years in a Range

Rule 1: A year is called leap year if it is divisible by 400. For example: 1600, 2000 etc leap year while 1500, 1700 are not leap year.
Rule 2: If year is not divisible by 400 as well as 100 but it is divisible by 4 then that year are also leap year.
For example: 2004, 2008, 1012 are leap year.

Enter the lowest year and highest year (lowest year,highest year). This forms the range. Check whether each year in this range is leap year or not.
For example if range is (2000,3000), then check whether the years from 2000 to 3000 ie 2000,2001,2002...3000 is a leap year using the rules. if a particular year is leap then print that year. Sample Output is

```
Enter the lowest year: 2000
Enter the highest year: 2011
Leap years in given range is: 2000 2004 2008
```

5 Power of a number

Read the input number and the power into two variables a,n.
Multiply the input number a, n times using loops.
Hint: Initialize result variable to 1.
Perform result=result*a; n times using loop

6 Count number of digits in a number

Read the input number, initialize a variable to count the no of digits to 0
Divide the number by 10 and increment counter variable. Do this step until number >0.
eg: number=123, counter=0.

number=number/10—-123/10 so number=12, counter=counter+1, so—counter=1

number=number/10—-12/10 so number=1, counter=counter+1, so—counter=2

number=number/10—-1/10 so number=0, counter=counter+1, so—counter=3

now number=0, so no need of dividing number again. the above 3 steps can be done in a single step using loop.

7 Calculate the sum of digits of a number and reverse it.

Initialize sum=0, reverse=0

remainder=number%10–remainder=3,sum=sum+remainder—sum=3, reverse=(reverse*10)+remainder–reverse=3, number=number/10—-123/10 so number=12,

remainder=number%10–remainder=2,sum=sum+remainder—sum=5, reverse=(reverse*10)+remainder–reverse=32,number=number/10—-12/10 so number=1,

remainder=number%10–remainder=1,sum=sum+remainder—sum=6, reverse=(reverse*10)+remainder–reverse=321 ,number=number/10—-1/10 so number=0,

now number=0, so no need of dividing number again. the above 3 steps can be done in a single step using loop.

8 Evaluate the sum of the series: 1-x^2+x^4-x^6.... upto N terms

Read value of x and number of terms N
For iterative computation of the sum, we will set
- value of term to 1 (the first term)
- sum to 0
- a counter variable k to 1
At the begining of each iteration, we will check
- if no of terms is less than or equal to N do the followiing
sum = sum + term;
term = (term * (-x*x))

9 Prime Number Checking

We will take a loop and divide number from 2 to number/2. If the number is not divisible by any of the numbers then we will print it as prime number.

Part II

Loop Control- Use while Loop

10 Check Whether Given Number is Palindrome or Not

A number is called palindrome number if it is remain same when its digits are reversed. For example 121 is palindrome number. When we will reverse its digit it will remain same number i.e. 121

11 Factorial of a Number

Factorial of number is defined as:
Factorial (n) = 1*2*3 * n
For example: Factorial of 5 = 1*2*3*4*5 = 120
Note: Factorial of zero = 1

12 Factors of a number

Divide the number with divisors from 2 to n/2.
If any of the division doesnot yield a remainder then print that divisor as a factor.

13 Check whether a number is Armstrong Number

those numbers which sum of the cube of its digits is equal to that number are known as armstrong numbers.
for example 153 since $1^3 + 5^3 + 3^3 = 1 + 125 + 9 = 153$ other armstrong numbers: 370,371,407 etc.
in general definition: those numbers which sum of its digits to power of number of its digits is equal to that number are known as armstrong numbers. example 1: 153
total digits in 153 is 3
and $1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$
example 2: 1634
total digits in 1634 is 4
and $1^4 + 6^4 + 3^4 + 4^4 = 1 + 1296 + 81 + 64 = 1634$
examples of armstrong numbers: 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474, 54748, 92727, 93084, 548834, 1741725

14 Check whether a number is Perfect Number

Perfect number is a positive number which sum of all positive divisors excluding that number is equal to that number. For example 6 is perfect number since divisor of 6 are 1, 2 and 3. Sum of its divisor is 1 + 2+ 3 =6

Note: 6 is the smallest perfect number.

Next perfect number is 28 since 1+ 2 + 4 + 7 + 14 = 28
Some more perfect numbers: 496, 8128

15 Check whether a number is Strong Number

A number is called strong number if sum of the factorial of its digit is equal to number itself.
For example: 145 since 1! + 4! + 5! = 1 + 24 + 120 = 145

16 Generate Fibanocci Series

F(n)=F(n-1)+F(n-2)
ie fibanocci series is 1,1,2,3,5,8,13..
We assume first two Fibonacci are 0 and 1 A series of numbers in which each sequent number is sum of its two previous numbers is known as Fibonacci series and each numbers are called Fibonacci numbers. So Fibonacci numbers is

Algorithm for Fibonacci series

Fn = Fn-2 + Fn-1
Example of Fibonacci series:
0 , 1 ,1 , 2 , 3 , 5 , 8 , 13 , 21 , 34 , 55 ...
5 is Fibonacci number since sum of its two previous number i.e. 2 and 3 is 5
8 is Fibonacci number since sum of its two previous number i.e. 3 and 5 is 8 and so on.

17 Prime factors of a number

Find the factors of the number.
Check whether each of these factors are prime.
If the factor is prime print it.

18 Display Floyd's triangle

Floyd's triangle is a right angled-triangle using the natural numbers. Examples of floyd's triangle:

Example 1:

```
1
2  3
4  5  6
7  8  9  10
```

Example 2:

```
1
2    3
4    5    6
7    8    9    10
```

11 12 13 14 15
16 17 18 19 20 21

You have to use nested loop ieloop inside loop

Part III

Loop Control- Use do while loop and switch case

19 Menu Driven Program to implement simple calculator

20 Multiplication table

Read the number upto which multiplication table has to be printed.
Print the multipication table in the following format:

```
Enter the starting value10
Enter the ending value10
            MULTIPLICATION TABLE
-----------------------------------------
   1    2    3    4    5    6    7    8    9   10
   2    4    6    8   10   12   14   16   18   20
   3    6    9   12   15   18   21   24   27   30
   4    8   12   16   20   24   28   32   36   40
   5   10   15   20   25   30   35   40   45   50
   6   12   18   24   30   36   42   48   54   60
   7   14   21   28   35   42   49   56   63   70
   8   16   24   32   40   48   56   64   72   80
   9   18   27   36   45   54   63   72   81   90
  10   20   30   40   50   60   70   80   90  100
```

21 Menu driven program to display : Armstrong Numbers, Prime Series, Fibonacci Series, Perfect Numbers and Strong Numbers in a range or limit

22 Menu driven program to display the following patterns:Even Number Pyramid , Pyramid of Multiplication Tables

Even Number Pyramid

```
2
2   4
2   4   6
2   4   6   8
```

Pyramid of multiplicationtable

```
0
0       1
0       2       4
0       3       6       9
0       4       8       12      16
0       5       10      15      20      25
0       6       12      18      24      30      36
```