Batch:C1_2	Roll No.:16010123032
Experiment / assignment / tutorial No. 1	
Grade: AA / AB / BB / BC / CC / CD /DD	
Signature of the Staff In-charge with date	

TITLE: Write a program for:
a. Program to find area and circumference of various Geometric shapes.
b. Program to calculate EMI (Equated Monthly Instalment) of loan amount if principal, rate of interest and time in years is given by the user.
$(E = (P.r.(1+r)^n) / ((1+r)^n - 1)$

AIM: Write a program for:
a. Program to find area and circumference of various Geometric shapes.
b. Program to calculate EMI (Equated Monthly Instalment) of loan amount if principal, rate of interest and time in years is given by the user.
$E = (P.r.(1+r)^n) / ((1+r)^n - 1)$

Expected OUTCOME of Experiment:
1. Find area and circumference of various Geometric shapes
2. To calculate EMI

Books/ Journals/ Websites referred:
1. Programming in ANSI C, E. Balagurusamy, 7 th Edition, 2016, McGraw-Hill Education, India.
2. Structured Programming Approach, Pradeep Dey and Manas Ghosh, 1 st Edition, 2016, Oxford University Press, India.
3. Let Us C, Yashwant Kanetkar, 15th Edition, 2016, BPB Publications, India.

Problem Definition:

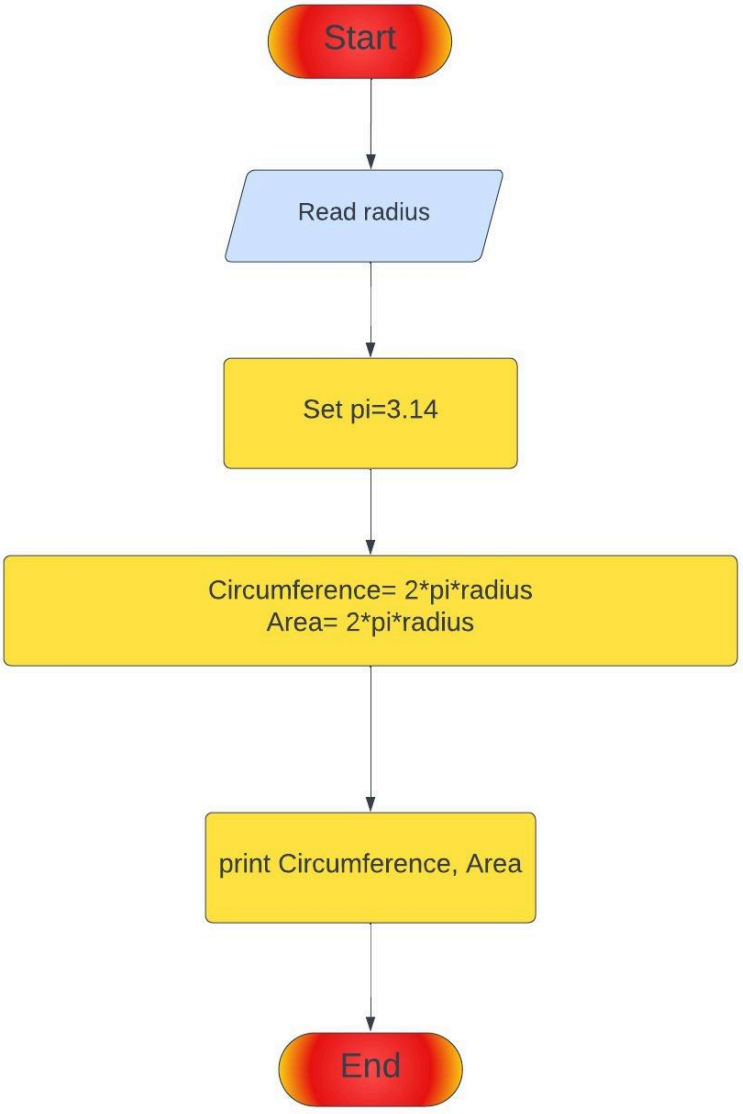
Problem 1: Area and Circumference of any shape(will be given by instructor) (example Circle)
Ask the user to enter the value of the radius of a circle. Put the values in the formula for finding area of a circle and circumference of a circle and print the outcome for area of a circle and circumference of a circle

Problem 2: Calculating EMI Ask the user to enter the value of principal amount, rate of interest and time (in years).Store the value in E and print the final monthly instalment E as an outcome.

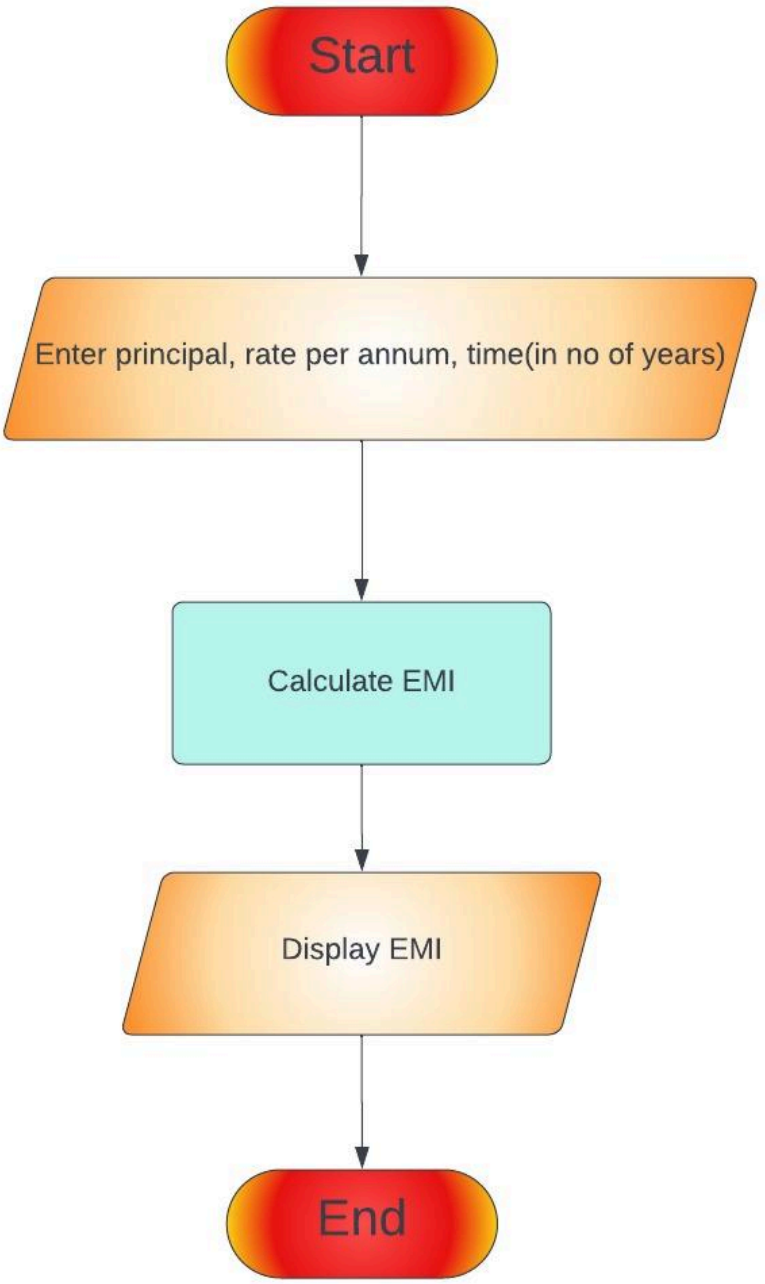
Formula to be used: $(E = (P.r.(1+r)^n) / ((1+r)^n - 1)$

Flowchart:

1)



2)


Start
Enter principal, rate per annum, time(in no of years)
Calculate EMI
Display EMI
End

Implementation details:

1)

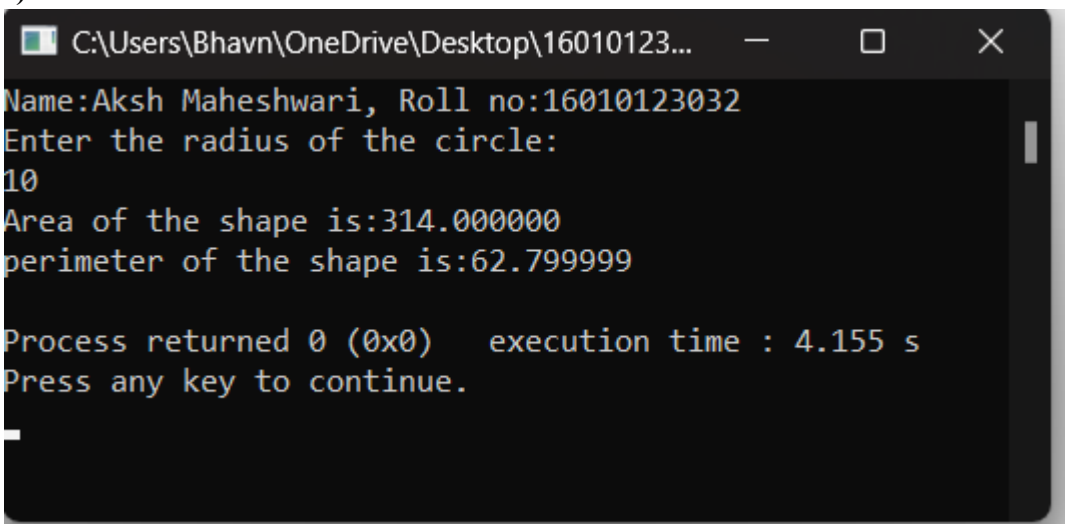
```c
#include <stdio.h>
int main ()
{
    printf("Name:Aksh Maheshwari, Roll no:16010123032\n");
    float r;
    printf("Enter the radius of the circle:\n");
    scanf("%f",&r);
    float area = 3.14*r*r;
    float peri = 2 * 3.14 * r;
    printf("Area of the shape is:%f\n",area);
    printf("perimeter of the shape is:%f\n",peri);
    return 0;



}
```

2)

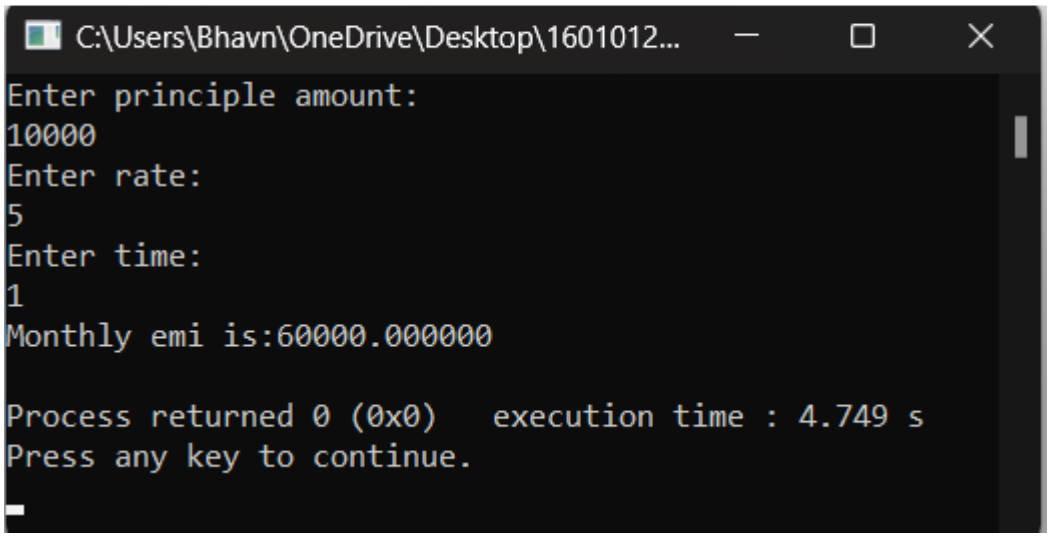
```c
#include <stdio.h>
#include <math.h>

int main ()
{
    float p;
    float r;
    float n;
    printf("Enter principle amount:\n");
    scanf("%f",&p);
    printf("Enter rate:\n");
    scanf("%f",&r);
    printf("Enter time:\n");
    scanf("%f",&n);
    //(E = (P.r.(1+r)^n) / ((1+r)^n - 1)
    float e = (p * r * pow(1 + r, n)) / (pow(1 + r, n) - 1);
    printf("Monthly emi is:%f\n",e);
    return 0;



}
```

Output(s):

1)


C:\Users\Bhavn\OneDrive\Desktop\16010123...
Name:Aksh Maheshwari, Roll no:16010123032
Enter the radius of the circle:
10
Area of the shape is:314.000000
perimeter of the shape is:62.799999
Process returned 0 (0x0) execution time : 4.155 s
Press any key to continue.

2)


C:\Users\Bhavn\OneDrive\Desktop\1601012...
Enter principle amount:
10000
Enter rate:
5
Enter time:
1
Monthly emi is:60000.000000
Process returned 0 (0x0) execution time : 4.749 s
Press any key to continue.

Conclusion:
I learnt basic data types in C and quite a few arithmetic operations and their use in C programming language

Post Lab Descriptive Questions

1) What are the basic data types in C?
Ans-

- int: Integer data type is used to store whole numbers.
- float: Float data type is used to store single-precision floating-point numbers.
- double: Double data type is used to store double-precision floating-point numbers.
- char: Char data type is used to store a single character

2) Write a table for Operator Precedence and Associativity.

Operator	*Description*	*Associativity*
() [] . ->>> ++ --	Parentheses or function call Brackets or array subscript Dot or Member selection operator Arrow operator Postfix increment/decrement	left to right
++ -- + - ! ~ (type) * & sizeof	Prefix increment/decrement Unary plus and minus not operator and bitwise complement type cast Indirection or dereference operator Address of operator Determine size in bytes	right to left
* / %	Multiplication, division and modulus	left to right
+ -	Addition and subtraction	left to right
<< >>	Bitwise left shift and right shift	left to right
< <= > >=	relational less than/less than equal to relational greater than/greater than or equal to	left to right
== !=	Relational equal to or not equal to	left to right
&&	Bitwise AND	left to right
^	Bitwise exclusive OR	left to right
|	Bitwise inclusive OR	left to right
&&	Logical AND	left to right
||	Logical OR	left to right
? :	Ternary operator	right to left
= += -= *= /= %= &= ^= |= <<= >>=	Assignment operator Addition/subtraction assignment Multiplication/division assignment Modulus and bitwise assignment Bitwise exclusive/inclusive OR assignment	right to left
,	comma operator	left to right

Date: _____

Signature of faculty in-charge