

1. Write a C function that takes a real number x and an integer n as its argument and returns the value of $\ln(1+x)$. It calculates the value of $\ln(1+x)$ using n -terms of the Taylor series

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots, \quad -1 < x < 1.$$

(Do not use *pow* function).

[6]

2. A given sequence a_n is defined by the recurrence relation

$$a_n = a_{n-1} + a_{n-2} + a_{n-3}, \quad n \geq 3, \quad a_0 = 0, \quad a_1 = 1, \quad a_2 = 2.$$

Write a non-recursive C function that accepts a non-negative integer n as its argument and returns the value of a_n .

[6]

3. Write a recursive C function that implements the algorithm of the previous question. What would be the total number of calls made to this recursive function for $n = 4$? Justify your answer.

[8]

4. Write down the output of the following program.

[7]

```
#include<stdio.h>
int myth(int);
int main()
{
int s;
s=myth(4);
return 0;
}
int myth(int n)
{
int t;
if(n>0 && n%2==0)
{
printf("%d Even\n",n);
t=myth(n/2);
return t+myth(n/4);
}
else if(n>0 && n%2 !=0)
{
printf("%d Odd\n",n);
return myth(2*n-2);
}
printf("Strange\n");
return 10;
}
```

5. Define a structure that describes a point in a two-dimensional Cartesian plane. Define another structure that describes a circle in a two-dimensional Cartesian plane having a centre and a radius. Write a C function which takes addresses of a point and a circle as its arguments and returns 1 if the point is within or on the circle and return 0 otherwise. [9]
6. Define a structure that describes a complex number. Write a C function that accepts two complex number as its argument and returns the sum of these complex numbers. [6]
7. Describe a structure that describes a given date having three components day, month and year. Write a function that accepts a date of December and returns the date of the next day. [8]
8. Copy the given code (omitting the comments) with modifications at appropriate places (see the comment parts of the program). Do not use extra variables. [10]

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
int main()
{
int *a,n,i;
double avg=0;
srand(time(NULL));
/*Read the value of n from the terminal*/
/*Create a dynamic array that can hold n integers*/
/*Initialized the array with random numbers between 10 and 20 both inclusive*/
/*Find the average of these random number and store the value in avg*/
/*Print the value of avg in the terminal*/
return 0;
}
```