Time:2 hours **Midsem: MTH409A** **Maximum Marks 50**

1. Rewrite the following code using **for** construct. What is the output of the code? [6]

```
int i,j=5;
i=1;
while(i+j<25)
{  if(j+i%4==11)break;
   if(i%2==0)
     { j /=2;
       i +=3;
       continue;
     }
   i--;
   j *=3;
 }
printf("%d %d\n",i,j);
```

2. An integer function $f(n)$ is defined as follows:

$$f(n) = \begin{cases} 4 & n = 0, \\ 1 & |n| = 2, \\ 7 & n = 7, \\ 0 & \text{otherwise.} \end{cases}$$

   Write a C program without using **if-else** construct that accepts an integer from the keyboard and prints the value of $f(n)$. [5]

3. Rewrite the following code using **while** construct. What is the output of the code? [6]

```
int i,j=1;
for(i=1,j++;i+j<12;i+=j++)
   { i--;
     j *=2;
   }
printf("%d %d\n",i,j);
```

4. Following is the Taylor series expansion of $\cos(x)$:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots.$$

   Write a C function that accepts a real value x and an integer n and returns the value of $\cos(x)$ using first n terms of the series. [7]

5. Write a C program that accepts a positive integer from the keyboard. If the input is invalid, it stops with appropriate message. For a valid input, it determines the first and last digits of the number. Further, it checks whether the first digit or the last digit is multiple of the other with appropriate message. [7]

6. Consider the following recursive *C* function:

```
int rec(int n)
{
if(n>=0 && n<=2)
  {
   return n;
  }
return rec(n-1)+2*rec(n-2)+rec(n-3);
}
```

Find the value return by **rec(5)**. Also, determine how many times the recursive function will be called for **rec(5)**. Justify your answer. [8]

7. Copy the given code (omitting the comments) with modifications at appropriate places (see the comment parts of the program). The function **dist** takes coordinates of two points and returns their distance. The function **Is_tri** takes three distances as arguments and returns 1 if they forms a triangle, zero otherwise. [11]

```
#include<stdio.h>
#include<math.h>
//Write the prototypes of functions dist and Is_tri here
int main()
{ int flag;
double xA,yA,xB,yB,xC,yC; //coordinates of points A, B and C
double AB,BC,CA; //AB=distance between points A and B, etc
//Read coordinates xA and yA of point A here
//Read coordinates xB and yB of point B here
//Read coordinates xC and yC of point C here
AB=dist(xA,yA,xB,yB);
BC=dist(xB,yB,xC,yC);
CA=dist(xC,yC,xA,yA);
flag=Is_tri(AB,BC,CA);
if(flag==1)
  { printf("A, B and C are vertices of a triangle\n");
  }
else
 { printf("A, B and C are NOT vertices of a triangle\n");
 }
return 0;
}
//Write the function dist here
//Write the function Is_tri here
```