Time:2:30 hours End Semester Examination: MTH409A Maximum Marks 65

1. Copy the given code (omitting the comments) and modify at appropriate places (see the comment parts of the program) so that the output of the code is 7 11 9. [8]

```
#include <stdio.h>
struct da1
{int a;
int *b;
};
struct da2
{int a;
struct da1 *b;
};
//Write the prototype of the function mystery here
int main()
{int i=5;
struct da2 p;
struct da1 q;
q.b=&i;
p.b=&q;
mystery(&p);
printf("%d %d %d\n",i,p.a,q.a);
return 0;
}
//write the details of the function mystery here
```

2. Copy the given code (omitting the comments) with modifications at appropriate places (see the comment parts of the program). Here m and n are the row and column dimensions of A and mt and nt are the row and column dimensions of A^T (transpose of A). After completion of the function matprint, the matrix is written row-wise in the terminal. After completion of the function Transp, mt and nt contain the row and column dimensions of A^T (transpose of A) and A now contains the components of A^T. (Do not use any extra array). [13]

```
#include <stdio.h>
#define N 10
//Write here the prototype of the function matprint & Transp
int main()
{   int A[N][N]={0},m,n,mt,nt,i, j;
    printf("Enter row and column dim. of the matrix: ");
    scanf("%d%d",&m, &n);
//Read here the element of the matrix row wise
    matprint(A,m,n);
    Transp(A,m,n,&mt,&nt);
    matprint(A,mt,nt);
    return 0;
}
//Write here the details of the function matprint
//Write here the details of the function Transp
```

3. Construct a structure variable **pt** which represents a point in xy-plane. A circle can be defined by its centre and radius. Using **pt**, construct a structure variable **cir** which represents a circle in the xy-plane. Write a C function which accepts addresses of two structure variables representing circle as arguments and returns the midpoint of the line segment joining their centres. [10]

4. Construct a binary search tree by inserting the values 13, 3, 4, 12, 14, 10, 1, 2, 11, 18 in that order starting from an entry tree. Write down the outcome of preorder traversal and postorder traversal on this binary search tree. [8]

5. Write a C function which accepts a string as argument and returns the last alphabet of the string. If the string has no alphabet, then the function returns '#'. For example, if the string argument is "How are you?", the function returns 'u'. (Do not use *strlen* function). [8]

6. Write down the output of the following C program. If a given output is unpredictable, write '#' in the corresponding place. [8]

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
int a[3][3]={{0,1},{2,3},{5,6}},**b,i,j;
b=(int **)calloc(3,sizeof(int *));
for(i=0;i<3;i++)
{
b[i]=(int *)calloc(3,sizeof(int));
}
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
b[i][j]=(i+1)*(j+1);
}
}
printf("%d  %d  %d  %d\n",*(*a+4),**a+4,*(*b+4),**b+4);
printf("%d  %d  %d  %d\n",*(a[2]-2),**(a+2),*(b[2]-2),**(b+2));
return 0;
}
```

7. Each node of a linked list consists of two components: an integer and an address of the next node. **Head** is variable that can hold the address of a node. Look at the schematic diagram below and implement it in a C program. Each node must be created using dynamic memory allocation. [10]