Time:2:30 hours End Semester Examination: MTH409A Maximum Marks 60

Note: *There are 7 questions.* Each question must begin on a *fresh page*, and prominently numbered. All parts of a given question *must* be answered in continuation, at one place. Number each page of your answer script and prepare a chart on the front page showing the page number and the question number attempted.

1. Write down the ouput of the following code: [8]

```
#include<stdio.h>
void write(char *);
int main()
{
char a[10]="LSG317";
write(a);
printf("\n");
printf("S=%d\n",a[2]+a[3]-'A'-a[4]);
return 0;
}
void write(char *p)
{
if(*p !='\0')
   {
    printf("%c",*p+1);
    write(p+1);
    printf("%c",*p);
   }
else
   printf("\n");
}
```

2. Copy the given code (omitting the comments) and modify at appropriate places (see the comment parts of the program). The function mswap swaps the two half of a character string about the middle. For example, if the string is "How-are", then the output is "are-How". On the other hand, if the string is "How-areY" then the output is "areYHow-". (Do not use *strlen* function and extra array variable). [8]

```
#include <stdio.h>
//Write here the prototype of the function mswap
int main()
{
char a[100]="How-are";
mswap(a);
printf("%s\n",a);
return 0;
}
//Write here the details of the function mswap
```

3. 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 8. [8]

```
#include <stdio.h>
struct ts
{int *a;
 int b;
};
typedef struct ts pst;
struct st
{pst a;
 int *b;
};
typedef struct st pnt;
//Write here the prototype of the function change
int main()
{int i=0,j=0;
pnt p;
pst q;
q.a=   ;       //Complete the assignment
p.a=   ;      //Complete the assignment
p.b=   ;      //Complete the assignment
change(&p);
printf("%d  %d\n",i,j);
return 0;
}
//Write here the details of the function change
```

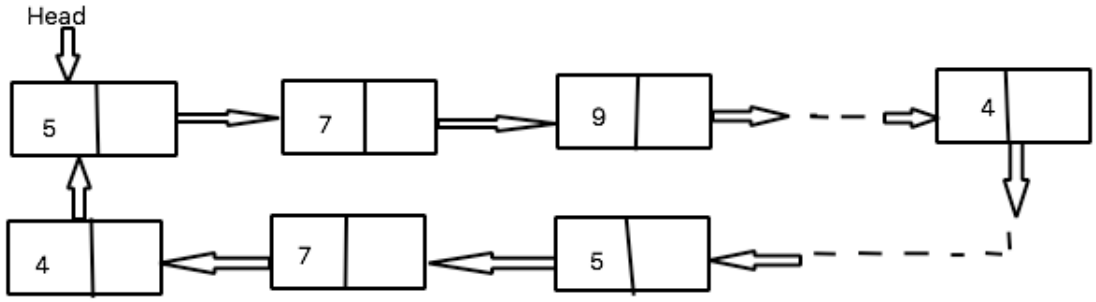
4. A rational number is of the form p/q where p and $q(\neq 0)$ are integers. If p and q are coprime, then the rational number is said to be in lowest form. For example, the rational number $-12/28$ reduces to $-3/7$ in lowest form. The program below calculates the lowest form of a rational number. Copy the given code (omitting the comments) and modify at appropriate places (see the comment parts of the program). [12]

```
#include<stdio.h>
//Define a structure here that represents a rational number
//Write here the prototype of the function lowest
int main()
{
//Declare here two variables a & b, both represent rational no.
//Read a rational number from keyboard into variable a
//If denominator is zero, stop with appropriate message
//If numerator is zero, print the lowest form as 0 and stop
//Call the function lowest here.
/*After call to lowest, b contains the rational number in lowest form*/
//Print the rational number in lowest form
return 0;
}
//Write the details of the function lowest
```

5. Construct a binary search tree by inserting the values 19, 18, 22, 21, 16, 17, 25, 24 in that order starting from an entry tree. Write down the outcome of preorder traversal and postorder traversal on this binary search tree. [6]

6. A perfect number is a positive integer that is equal to the sum of its positive divisors excluding the number itself. For example, 28 is a perfect number since $28 = 1 + 2 + 4 + 7 + 14$ and $1, 2, 4, 7, 14$ are its proper positive divisors. Write a C function which accepts an integer as argument and returns 1 if the integer is a perfect number, otherwise, it returns 0. (You may assume that the number is positive). [6]

7. Each node of a linked list (schematic diagram shown below) consists of two components: an integer and an address of a node. Here, **Head** *is a variable that holds the address of the first node*. Each node is created using dynamic memory allocation. After the call to the insert function, a new node with *key=8* is added between the first node (i.e. the node pointed by Head) and the last node. After the call to the delete function, the second node in the list is deleted. Copy the given code (omitting the comments) and fill up the details at the commented place marked by (1), (2), (3) and (4). (You may assume that the list has at least two nodes before call to each function). [12]




```
#include<stdio.h>
#include<stdlib.h>
struct nd
{
int key;
struct nd *next;
};
typedef struct nd node;
//(1) Write the prototype of function insert here
//(2) Write the prototype of function delete here
int main()
{
node *Head;
/*Linked list is created here for you &
Head has address of the starting node*/
insert(Head,8);
delete(Head);
return 0;
}
//(3) Write the details of the function insert here
//(4) Write the details of the function delete here
```