



**Question [1] (25 marks)**

a) **What is the meaning of: data structures, stacks and queues?**  
 (Explain your answer by details.)

**Data Structure:** is a way of collecting and organizing data in such a way that we can perform operations on these data in an effective way. Data Structures is about rendering data elements in terms of some relationship, for better organization and storage.

**Stack :** is an abstract data type with a bounded (predefined) capacity. It is a simple data structure that allows adding and removing elements in a particular order. Every time an element is added, it goes on the top of the stack, the only element that can be removed is the element that was at the top of the stack, just like a pile of objects. Stack is a **LIFO** structure. (Last in First out).

**Queue:** is also an abstract data type or a linear data structure, in which the first element is inserted from one end called **REAR** (also called tail), and the deletion of existing element takes place from the other end called as **FRONT**(also called head). This makes queue as FIFO data structure, which means that element inserted first will also be removed first.

b) **Find the error of the following program and correct it (make a table):**

#	Code	√ or X	Error	Correction
1	#include <stdio.h>	√		
2	void main()	√		
3	{	√		
4	float x,y,z,avg;	√		
5	printf("Enter 3 score values separated by space:");	√		
6	scanf("%f %f %f",&x,&y,&z);	X	x	&x
7	avg=(x+y+z)/3;	√		
8	if(avg>=90);	X	;	
9	printf("Grade A");	√		
10	else if((avg>=70) && (avg<90))	√		
11	printf("Grade B");	√		
12	else if((avg>=50) & (avg<70))	X	&	&&
13	printf("Grade C");	√		
14	else if(avg<50)	√		
15	printf("Grade <u>F</u> ");	X	F	F"
16	else	√		
17	printf("Invalid")	X	)	);
18	printf("\n");	√		
19	}	√		



**Question [2]: (20 marks)**

a) What is the output of the following C program?

```
#include <stdio.h>
void main()
{
    int num=5665, tNum, cnt;
    cnt=0;
    tNum=num;
    while(tNum>0)
    {
        cnt++;
        tNum/=10;
    }
    printf("The counter is: %d \n",cnt);
}
```

**The Output is:**

The counter is: 4

b) What is the output of the following C program?

```
#include<stdio.h>
int a[5];
void main(void)
{
    static int b[5];
    int c[5];
    int i;
    for (i=0; i<5; i++)
        printf("a[%d]=%d\t b[%d]=%d\t c[%d]=%d\n", i, a[i], i, b[i], i, c[i]);
}
```

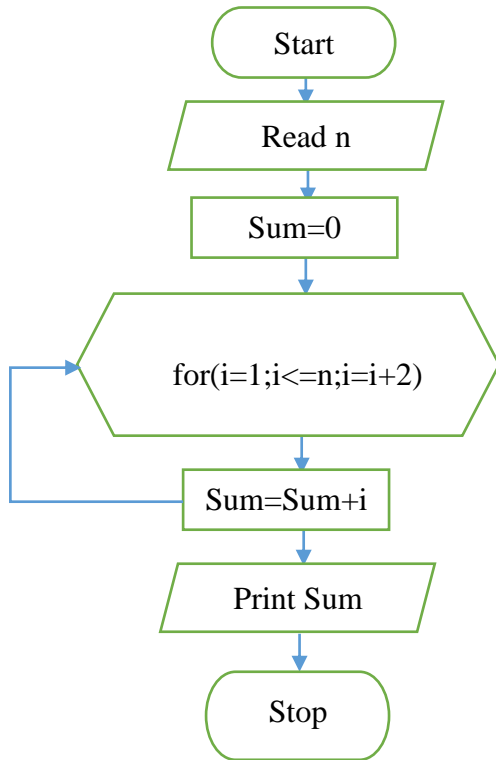
**The Output is:**

a[0]=0	b[0]=0	c[0]=125
a[1]=0	b[1]=0	c[1]=-13
a[2]=0	b[2]=0	c[2]=5431
a[3]=0	b[3]=0	c[3]=4573
a[4]=0	b[4]=0	c[4]=98

**Question [3]: (40 marks)**

a) Draw a flowchart; write an algorithm and writ a C program to find the sum of odd integer numbers from 0 to N.

**Algorithm:** Step 1: Input n  
Step 2: sum=0  
Step 3: for i=1 to i<=n, do  
          sum=sum+i  
          i=i+2  
Step 4: print sum



```
#include <stdio.h>
void main()
{
    int i,n,sum=0;
    printf("Enter the upper limit N to find sum: ");
    scanf("%d",&n);
    for(i=1;i<=n;i=i+2)
        sum=sum+i;
    printf("Sum of all odd numbers from 1 to %d is %d\n", n, sum);
}
```

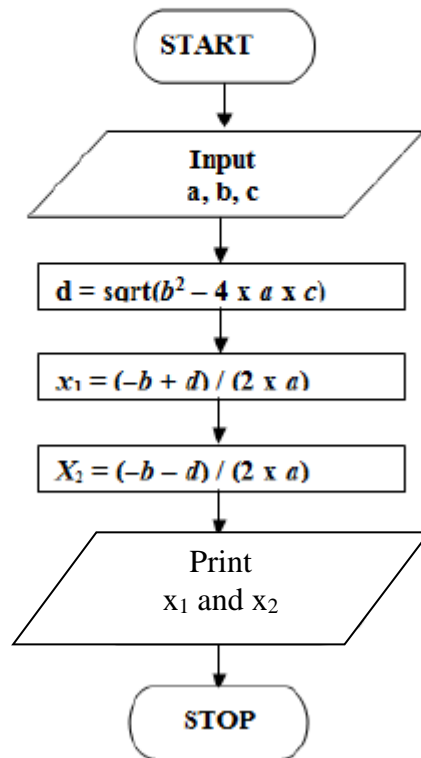
b) Draw a flowchart, write an algorithm and write a C program to compute the real roots of the equation:  $ax^2 + bx + c = 0$ .

**Algorithm:**

- Step 1: Input a, b, c
- Step 2:  $d = \text{sqrt}(b^2 - 4 \times a \times c)$
- Step 3:  $x_1 = (-b + d) / (2 \times a)$
- Step 4:  $x_2 = (-b - d) / (2 \times a)$
- Step 5: Print  $x_1$  and  $x_2$



**Flowchart:**



**C Program:**

```
#include <stdio.h>
#include <math.h>
void main()
{
    float a,b,c,d,x1,x2;
    printf("Enter values of a b c separated by space:");
    scanf("%f %f %f",&a,&b,&c);
    d=b*b-4.0*a*c;
    x1=(-b+sqrt(d))/(2.0*a);
    x2=(-b-sqrt(d))/(2.0*a);
    printf("The equation has two real roots:x1=%.2f, x2=%.2f\n",x1,x2);
}
```

**Other C program:**

```
#include <stdio.h>
#include <math.h>
void main()
{
    float a,b,c,delta;
    printf("Enter values of a b c separated by space:");
    scanf("%f %f %f",&a,&b,&c);
    if(a==0 && b==0)
```



```
printf("No roots");
else if(a==0)
printf("The equation has only one root: x1=%.2f",-c/b);
else
{
delta=b*b-4.0*a*c;
if(delta<0)
printf("No real roots");
else
printf("The equation has two real roots:x1=%.2f, x2=%.2f",(-b+sqrt(delta))/(2.0*a),
(-b-sqrt(delta))/(2.0*a));
}
printf("\n");
}
```

c) Write a C program to get the multiplication of two matrices A and B:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

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

d) Write a C program that sorts the elements of an array in ascending order using the bubble sort algorithm.

```
#include<stdio.h>
void main()
{
int num[5]={9,12,7,13,5};
```

