

Attempt each question on a fresh page

1. Define a structure that describes a complex number. Write a C function that accepts a complex number as its argument and returns the magnitude of the complex number. [6]
2. Write down the output of the following program. Find the total number of calls made to the function **recur**. Justify your answer. [10]

```
#include<stdio.h>
void recur(char[],int);
int main()
{
int n=0;
char a[20]="Computer Prog";
while(a[n] !='\0')
{
n++;
}
printf("n=%d\n",n);
recur(a,n);
return 0;
}
void recur(char a[],int n)
{
if(n<=2)
{
printf("\n");
}
else
{
printf("%c-",a[1]);
printf("%d x ",n);
recur(a+2,n-2);
}
}
```

For each problem given below, copy the given code (omitting the comments) with modifications at appropriate places (see the comment parts of the program).

3. The function **chcv** replaces the consonants of a string by * and vowels by !. For example, the output of the program is *!***!*!* **!*.* [7]

```
#include<stdio.h>
//Write the function prototype here
int main()
{
char a[20]="Computer Prog";
chcv(a+5);
printf("%s\n",a);
return 0;
}
//Write the function here
```

4. The program computes the number of values (**nc**) less than the average value of a set of **n** integer values (stored in **a**) using a function **nvalav**. [10]

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
//Write the function prototype here
int main()
{
int *a,i,n,nc;
srand(time(NULL));
//Read the value of n from the terminal
//Allocate space for n integer elements and assign the address to a
//Initialized the array with random numbers between -50 and 50
nc=nvalav(a,n);
printf("Number of value less than average value is %d\n",nc);
return 0;
}
//Write the function here
```

5. After call to the function **distbp**, the variable **s** contains the distance between two points **p** and **q** and the point **r** contains the midpoint. [10]

```
#include<stdio.h>
#include<math.h>
typedef struct
{
double x,y;
}point;
//Write the function prototype here
int main()
{
point p,q,r;
double s;
//Assign point (2.0,3.0) to p
//Assign point (4.0,5.0) to q
r=distbp(p,&q,&s);
printf("Distance between the points=%lf\n",s);
//print the midpoint r using two decimal places
return 0;
}
//Write the function here
```

6. The **recursive** function **recsum** returns the sum $\sum_{i=1}^n a_i$ of the array elements, where n is the array size. For example, the output of the program is **Sum=45**. [7]

```
#include<stdio.h>
//Write the function prototype here
int main()
{
int s,a[10]={1,3,2,5,4,7,6,9,0,8};
s=recsum(a,10);
printf("Sum=%d\n",s);
return 0;
}
//Write the function here
```

7. The program computes the value of

$$\tanh^{-1} x = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{2n+1}, \quad |x| < 1$$

using a function **valitanh**. After call to the function, flag will have value **0** if x is outside the range. Otherwise, flag will have value **1** and val will have value of $\tanh^{-1} x$ using **20** terms of the Taylor series. (*Do not* use **pow** function). [10]

```
#include<stdio.h>
//Write the function prototype here
int main()
{
int flag;
double x,val;
//Read the value of x from the terminal
flag=valitanh(x,&val);
if(flag==0)
    printf("x is outside the range\n");
else
    printf("Value is %0.3lf\n",val);
return 0;
}
//Write the function here
```