ASSIGNMENT-11

Write a C Program using structures for the following problem statements A. Array Manipulations using Pointers :-

A. Array Manipulations using Pointers

One Dimensional Array:

1. Find Mean, Median, Mode, Variance, Standard Deviation, and Range of 'n' elements in an array

```
#include<stdio.h>
int main()
{
int arr[50],*p;
p=arr;
printf("Enter size : ");
int size;
scanf("%d",&size);
printf("Enter %d elements : ",size);
for(int i=0; i<size; i++)
scanf("%d",p+i);
// code for finding mean...
float mean,sum=0.0;
for(int i=0; i<size; i++)
sum+=*(p+i);
mean=sum/size;
printf("Mean = %.2f",mean);
// code for sorting the array...
for(int round=1; round<size; round++)
{
for(int i=0; i<size-round; i++)
 sum+=*(p+i);
mean=sum/size;
printf("Mean = %.2f",mean);
// code for sorting the array...
for(int round=1; round<size; round++)
{
for(int i=0; i<size-round; i++)
{
if(*(p+i)>*(p+i+1))
{
int temp=*(p+i);
*(p+i)=*(p+i+1);
*(p+i+1)=temp;
}
}
}
// code to find the median...
float median;
if(size%2)
```

```
median=*(p+size/2);
else
median=(*(p+size/2)+*(p+(size/2)-1))/2.0;
printf("\nMedian = %.2f",median);
// code to find the mode...
int C=0,mode;
for(int i=0; i<size; i++)
{
int count=0;
for(int j=0; j<size; j++)
{
if(*(p+i)==*(p+j))
count++;
}
if(count>C)
{
C=count;
mode=*(p+i);
}
}
if(C==1)
printf("\nNo Mode.");
else
printf("\nMode = %d",mode);
// code to find the variance...
sum=0.0;
for(int i=0; i<size; i++)
sum+=(*(p+i)-mean)*(*(p+i)-mean);
printf("\nVariance = %.2f",sum/size);
// code to find the standard deviation...
printf("\nStandard Deviation = %.2f",sqrt(sum/size));
// code to find the range...
printf("\nRange = %d",*(p+size-1)-*(p));
getch();
return 0;
}
```

Output:-

```
Enter size : 3
Enter 3 elements : 34 67 89
Mean = 63.33
Median = 67.00
No Mode.
Variance = 510.89
Standard Deviation = 22.60

Range = 55
```

2. Sort the 'n' elements of an array in Descending order

```
#include<stdio.h>
```

```
int main()

{
   int *ptr,arr[5],i,j,temp=0;
   ptr = arr;
   for(i=0;i<5;i++){
      printf("Enter the elements- ");
      scanf("%d",ptr+i);
   }

   for(i=0;i<5;i++)
   {
      for(j=0;j<5;j++)
      {
        if(*(ptr + i) > *(ptr + j))
        {
           temp = *(ptr + j);
           *(ptr+j) = *(ptr + i);
           *(ptr+i) = temp;
        }
      }
   }
   printf("Elements in Descending order :: ");
   for(i=0;i<5;i++)
   {
      printf("%d ",*(ptr + i));
   }
}
```

Output:-

```
Enter the elements- 2
Enter the elements- 5
Enter the elements- 6
Enter the elements- 1
Enter the elements- 8
Elements in Descending order :: 8 6 5 2 1
```

3. Find the second largest and smallest element in an array

```
#include<stdio.h>
int main()
{
   int *ptr,arr[5],i,j,temp=0;
   ptr = arr;
   for(i=0;i<5;i++)
   {
      printf("Enter a number- ");
      scanf("%d",ptr+i);
   }

   for(i=0;i<5;i++)
   {
      for(j=0;j<5;j++)
      {
        if(*(ptr + i) > *(ptr + j))
        {
```

```
            temp = *(ptr + j);
            *(ptr+j) = *(ptr + i);
            *(ptr+i) = temp;
         }
       }
    }
  printf("Second largest number: %d\n",*(ptr+1));
  printf("Smallest number: %d",*(ptr+4));
}
```

Output:-

```
Enter a number- 4
Enter a number- 7
Enter a number- 8
Enter a number- 2
Enter a number- 1
Second largest number: 7
Smallest number: 1
```

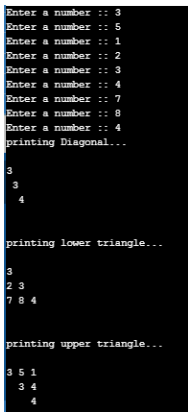
Two Dimensional Array:

4. Print the leading diagonal, upper triangular and lower triangular elements of mxm array

```
#include<stdio.h>
int main()
{
int arr[3][3],i,j,s=0,k=0;
int (*ptr)[3];
ptr = arr;

for(i=0;i<3;i++)
{
   for(j=0;j<3;j++)
   {
      printf("Enter a number :: ");
     scanf("%d",(*(ptr+i)+j));
   }
}
printf("printing Diagonal...\n\n");
for(i=0;i<3;i++)
{
   j = i ;
   k=0;
      while(k<s)
      {
         printf(" ");
         k++;
      }
      printf("%d ",*(*(ptr+i)+j));

    s++;
    printf("\n");
   }

printf("\n\nprinting lower triangle...\n\n");
for(i=0;i<3;i++)
{
```

```
   for(j=0;j<=i;j++)
   {
      printf("%d ",*(*(ptr+i)+j));
   }
   printf("\n");
}
s=0;
printf("\n\nprinting upper triangle...\n\n");
for(i=0;i<3;i++)
{
   k=0;
   for(j=i;j<3;j++)
   {

      while(k<s)
      {
         printf(" ");
         k++;
      }
      printf("%d ",*(*(ptr+i)+j));
   }
   s+=2;
   printf("\n");
}

}
```

Output:-

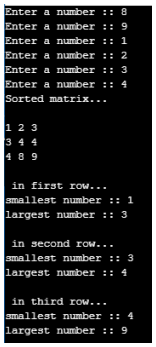


5. Find the maximum & minimum element in each row and each coloumn of mxm array

```
#include<stdio.h>
int main()
{
int arr[3][3],i,j,k,l,temp=0;
int (*ptr)[3];

ptr = arr;

for(i=0;i<3;i++)
{
   for(j=0;j<3;j++)
```

```
   {
      printf("Enter a number :: ");
     scanf("%d",(*(ptr+i)+j));
   }
}

for(i=0;i<3;i++)
{
   for(j=0;j<3;j++)
   {
    for(k=0;k<3;k++)
    {
       for(l=0;l<3;l++)
       {

          if(*(*(ptr+i)+j) < *(*(ptr+k)+l))
          {

             temp = *(*(ptr+k)+l);
             *(*(ptr+k)+l) = *(*(ptr+i)+j);
             *(*(ptr+i)+j) = temp;
          }
       }
    }
   }
}
printf("Sorted matrix...\n\n");
for(i=0;i<3;i++){
   for(j=0;j<3;j++){
      printf("%d ",*(*(ptr+i)+j));
   }
   printf("\n");
}

printf("\n in first row...\n");
printf("smallest number :: %d\nlargest number :: %d\n",*(*(ptr+0)+0),*(*(ptr+0)+2));

printf("\n in second row...\n");
printf("smallest number :: %d\nlargest number :: %d\n",*(*(ptr+1)+0),*(*(ptr+1)+2));

printf("\n in third row...\n");
printf("smallest number :: %d\nlargest number :: %d",*(*(ptr+2)+0),*(*(ptr+2)+2));

}
```

Output:-



6. Perform matrix multiplication between two mxm array

```
#include<stdio.h>
int main()
{
int arr[3][3],arr1[3][3],mul[3][3],i,j,k,l,temp=0;
int (*ptr)[3],(*ptr1)[3],(*res)[3];

ptr = arr;
ptr1 = arr1;
res = mul;
printf("Enter elements in first array... \n");
for(i=0;i<3;i++)
{
   for(j=0;j<3;j++)
   {
      printf("Enter a number :: ");
     scanf("%d",(*(ptr+i)+j));
   }
}

printf("Enter elements in second array... \n");
for(i=0;i<3;i++){
   for(j=0;j<3;j++){
      printf("Enter a number :: ");
     scanf("%d",(*(ptr1+i)+j));
   }
}

for(i=0;i<3;i++){
   for(j=0;j<3;j++){
      *(*(res+i)+j) = 0;
    for(k=0;k<3;k++){
       *(*(res+i)+j) += *(*(ptr+i)+k) * *(*(ptr+k)+j);
        }
    }
   }

printf("Matrix multiplication...\n\n");
for(i=0;i<3;i++){
   for(j=0;j<3;j++){
```

```
      printf("%d ",*(*(res+i)+j));
    }
    printf("\n");
}


 return 0;
}
```

Output:-

```
Enter a number :: 4
Enter a number :: 1
Enter a number :: 5
Enter a number :: 4
Enter a number :: 3
Enter a number :: 2
Enter a number :: 3
Enter elements in second array...
Enter a number :: 3
Enter a number :: 2
Enter a number :: 3
Enter a number :: 4
Enter a number :: 2
Enter a number :: 4
Enter a number :: 1
Enter a number :: 5
Enter a number :: 3
Matrix multiplication...

23 24 32
20 35 36
20 22 29
```

B. String Manipulations using Pointers

7. Write a C Program to convert:-

a. Upper case to Lower case

```
#include<stdio.h>
int main()
{
int i=0;
char *ptr,ch[15];

ptr = ch;

printf("Enter a string in Upper case: ");
gets(ptr);

printf("String in lower case: ");
while(*(ptr+i)!= '\0')
{
   printf("%c",*(ptr+i)+32);
   i++;
}
 return 0;
}
```

Output:-

```
Enter a string in Upper case: COMPUTER
String in lower case: computer
```

b. Lower case to Upper case

```
#include<stdio.h>
int main()
{
int i=0;
char *ptr,ch[15];
```

```
ptr = ch;

printf("Enter a string in lower case: ");
gets(ptr);

printf("String in upper case: ");
while(*(ptr+i)!= '\0')
{
   printf("%c",*(ptr+i)-32);
   i++;
}
 return 0;
}
```

Output:-

```
Enter a string in lower case: computer
String in upper case: COMPUTER
```

c. Toggle case

```
#include<stdio.h>
int main()
{
int i=0;
char *ptr,ch[15];
ptr = ch;
printf("Enter a string :: ");
gets(ptr);
printf("String in Toggle case :: ");
while(*(ptr+i)!= '\0'){
   if(*(ptr+i)>=65 && *(ptr+i)<=90)
   {
      printf("%c",*(ptr+i)+32);
   }
   else
   {
      printf("%c",*(ptr+i)-32);
   }
   i++;
}
 return 0;
}
```

Output:-

```
Enter a string :: COMpuTER
String in Toggle case :: comPUter
```

8. Write a C Program to read 2 string constants into a and b. Compare whether they are equal or not. if not, join them together. Then copy the contents of a to the variable c. At the end of the program, print the contents of all three variables and their length. (With and Without String Handling Functions).

```
#include<stdio.h>
int main()
{
char str[20],str1[20],str3[20],*a,*b,*c;
```

```
a=str;
b=str1;
c=str3;
printf("Enter a string : ");
gets(a);
printf("Enter another string : ");
gets(b);
printf("Using string handling functions...\n");
if(strcmp(a,b))

strcpy(c,a);
strcat(a,b);
}
printf("a = %s and size = %d\nb = %s and size = %d\nc = %s and size =
%d\n",a,strlen(a),b,strlen(b),c,strlen(c));
printf("\nWithout using string handling functions...");
a=str;
b=str1;
c=str3;
printf("\nEnter a string : ");
gets(a);
printf("Enter another string : ");
gets(b);
int i;
for(i=0; *(a+i)!='\0'; i++)
{
if(*(a+i) != *(b+i))
{
i=-1;
break;
}
}
if(i=-1)

{
int l1=strlen(a);
int l2=strlen(b);
for(i=0; *(a+i)!='\0'; i++)
*(c+i)=*(a+i);
*(a+l1)=32;
int j=0;

for(int i=l1+1; i<=l1+l2; i++,j++)
*(a+i)=*(b+j);
}
printf("\na = %s and size = %d\nb = %s and size = %d\nc = %s and size =
%d\n",a,strlen(a),b,strlen(b),c,strlen(c));
}
```

Output:-

```
Enter a string : its devil
Enter another string : data structure
```

```
Using string handling functions...
a = its devil data structure and size = 27
b = data structure and size = 14
c = its devil and size = 13
Without using string handling functions...
Enter a string : welcome
Enter another string : programming
a = welcome programming structure and size = 27
b = programming and size = 11
c = welcomedevvil and size = 13
```

9. Write a C program to read a string and prints if it is a palindrome or not.

```
#include<stdio.h>

int main()
{
int c=0,i=0,p=0;
char *ptr,ch[15];

ptr = ch;

printf("Enter a string: ");
gets(ptr);

while(*(ptr+i) != '\0')
{
 c++;
 i++;
}
c = c-1;

i=0;
while(i<=c)
{
 if(*(ptr+i) == *(ptr+c-i))
 {
    p++;
 }
 i++;
}
c = c+1;

if(p == c)
{
   printf("its a palindrome string");
}
else
{
   printf("its not a palindrome string");
```

```
}
 return 0;
}
```

Output:-

```
Enter a string: madam
its a palindrome string
```

C. Functions using Pointers:-

10. Check Prime and Armstrong Number by making function

```
#include<stdio.h>
void checkPrime();
void checkArmstrong();

int main(){
  int num;

  printf("enter a number :: ");
  scanf("%d",&num);
  checkPrime(&num);
 checkArmstrong(&num);

 return 0;
}

void checkPrime(int *ptr){
   int i=1,c=0;
   while(i<=*ptr){
    if(*ptr % i == 0){
       c++;
    }
    i++;
   }
   if(c == 2){
      printf("%d is a prime number\n\n",*ptr);
   }
   else{
      printf("%d is not a prime number\n\n",*ptr);
   }
}

void checkArmstrong(int *ptr){
    int copy,num,d=0,sum=0;
    num = *ptr;
    copy = *ptr;

   while(num > 0){
      d = num % 10;
      sum += d * d * d;
      num /= 10;
   }
   if(sum == copy)
```

```
    printf("%d is a armstrong number\n\n",copy);
    else
    printf("%d is not a armstrong number\n\n",copy);
}
```

Output:-

```
enter a number :: 23
23 is a prime number

23 is not a armstrong number
```

11. Reverse a sentence using String Functions

```
#include<stdio.h>
void reverse();
int main()
{
  char str[30];
  printf("Enter a sentence :: ");
  gets(str);
  printf("Reverse Sentence :: ");
  reverse(str);

 return 0;
}

void reverse(char *ptr)
{
    int i=0,c=0;
    while(*(ptr+i) != '\0')
    {
      c++;
       i++;
    }
    i=0;
    c = c-1;

    while(c>=i)
    {
      printf("%c",*(ptr+c));
      c--;
    }

}
```

Output:-

```
Enter a sentence :: program4
Reverse Sentence :: 4margorp
```

12. Calculate the power of a number using recursion

```
#include<stdio.h>
int res=1,s;
int power(int *n,int *p)
{
 if(*p == 0)
 {
```

```
    return 1;
 }
 else
 {
   s = *p-1;
    return(*n * power(n,&s));
 }

 }

int main()
{
int num,pow,mul;

printf("Enter number :: ");
scanf("%d",&num);
printf("Enter power :: ");
scanf("%d",&pow);
 mul = power(&num,&pow);
 printf("%d ^ %d = %d",num,pow,mul);
   return 0;
}
```

Output:-

```
Enter number :: 2
Enter power :: 3
2 ^ 3 = 8
```

D. Structures using Pointers:-

13. Store Information (name, roll and marks) of a Student Using Structure

```
#include<stdio.h>
 int main()
 {
 struct student
 {
   char name[20];
   int rno,marks;
 };
 struct student std;
 struct student *ptr=NULL;
 ptr=&std;
 printf("enter name\n");
 scanf("%s",ptr->name);
 printf("enter roll no\n");
 scanf("%d",&ptr->rno);
 printf("enter marks\n");
 scanf("%d",&ptr->marks);
 printf("printing the details of student:\n");
 printf("name:%s\n",ptr->name);
 printf("roll no:%d\n",ptr->rno);
 printf("marks:%d\n",ptr->marks);
 return 0;
 }
```

Output:-

```
enter name
kriti
enter roll no
21
enter marks
89
printing the details of student:
name:kriti
roll no:21
marks:89
```

14. Add Two Complex Numbers by Passing Structure to a Function

```
#include <stdio.h>
typedef struct complex
{
float real;
float imag;
} complex;
complex add(complex n1,complex n2);
main()
{
complex n1, n2, temp;
printf("For 1st complex number \n");
printf("Enter real and imaginary part respectively:\n");
scanf("%f %f", &n1.real, &n1.imag);
printf("\nFor 2nd complex number \n");
printf("Enter real and imaginary part respectively:\n");
scanf("%f %f", &n2.real, &n2.imag);
temp = add(n1, n2);
printf("Sum = %.1f + %.1fi", temp.real, temp.imag);
}
complex add(complex n1, complex n2)
{
complex temp;
temp.real = n1.real + n2.real;
temp.imag = n1.imag + n2.imag;
return(temp);
}
```

Output:-

```
For 1st complex number
Enter real and imaginary part respectively:
23.2
11.5

For 2nd complex number
Enter real and imaginary part respectively:
15.20
20.5
Sum = 38.4 + 32.0i
```

15. Store & Retrieve Information, Calculate Total, Average and Rank of 10 Students Using Structure

```
#include<stdio.h>
typedef struct

char name[20];
int m1;
int m2;
int m3;
}students;
int main()
```

```
{
students s[10],*p;
p=s;
printf("Enter details of 10 students : \n");
for(int i=0; i<10; i++)
{
printf("Enter details of student %d :\n",i+1);
printf("Name : ");
fflush(stdin);
gets(p[i].name);
printf("Enter mark 1 : ");
scanf("%d",&(p+i)->m1);
printf("Enter mark 2 : ");
scanf("%d",&(p+i)->m2);
printf("Enter mark 3 : ");
scanf("%d",&(p+i)->m3);

for(int i=0; i<10; i++)
{
int sum=0;
sum=(p+i)->m1+(p+i)->m2+(p+i)->m3;
printf("%s\nSum of marks : %d, Average : %f",(p+i)->name,sum,sum/3.0);
float per=(sum*100)/600;
if(per>=60)
printf(" Rank 1");

else if(per>=50 && per<60)
printf(" Rank 2");
else if(per>=30)
printf(" Rank 3");
else
printf(" Fail");
printf("\n");
}
}
```

Output:-

```
Enter details of 10 students :
Enter details of student 1 :
Name : Kriti
Enter mark 1 : 67
Enter mark 2 : 54
Enter mark 3 : 44
Enter details of student 2 :
Name : Enter mark 1 : Sonali

Enter mark 2 : Enter mark 3 : Enter details of student 3 :
Name : Enter mark 1 : 23
Enter mark 2 : 45
Enter mark 3 : 67
Enter details of student 4 :

Name : Enter mark 1 : Sonali
Enter mark 2 : Enter mark 3 : Enter details of student 5 :
```

```
Name : Enter mark 1 : 33
Enter mark 2 : 22
Enter mark 3 : 89
Enter details of student 6 :
Name : Enter mark 1 :
```