Name	Debjit Ghosal
UID no.	2023300065
Experiment No.	9

AIM:	**Demonstrate the use of pointers to solve a given problem**
Program 1	
PROBLEM STATEMENT :	Write a program to find smallest and largest element in an array using pointers
PSEUDO CODE :	readArr(n, arr) for i from 0 to n - 1 input arr[i] procedure minMaxArr(n, arr) min = arr[0] max = arr[0] for i from 1 to n - 1 if arr[i] < min min = arr[i] else if arr[i] > max max = arr[i] output "Smallest element in array is", min, "and largest element in array is", max procedure main() output "Enter number of elements of the array: " input n declare arr[n] output "Enter elements: " readArr(n, arr) minMaxArr(n, arr) return 0

PROGRAM:

```
#include<stdio.h>
#include<string.h>
void read_array(int*a,int n)
        {
        for(int i=0;i<n;i++)
        scanf("%d",&a[i]);
        }

void print_array(int*a,int n)
        {
        for(int i=0;i<n;i++)
                {
                printf("%d",a[i]);
                }
        }

void find_large_small(int *a,int n)
        {
        int *pmax=a;
        int *pmin=a;
   for(int i=1;i<n;i++)
   {
   if(*pmax<*(a+i))
        pmax=a+i;
        if(*pmin>*(a+i))
        pmin=a+i;
   }
   printf("\nThe minimum = %d",*pmin);
        printf("\n The maximum =%d \n",*pmax);

        }

int main()
{
        int pmin,pmax,n;
        printf("Enter the number of elements : ");
        scanf("%d",&n);

        int a[n];
        read_array(a,n);
```

```
        find_large_small(a,n);

return 0;
}
```

RESULT:

```
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$ gcc largest_smallest_pointer.c
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$ ./a.out
Enter the number of elements : 5
45
78
32
64
22
The array before Reverse : 4578326422
The array after Reverse : 2264327845
The minimum = 22
 The maximum =78
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$
```

Program 2	
PROBLEM STATEMENT :	Write a program to reverse all elements in the array using Pointers
PSEUDO CODE:	procedure readArr(n, arr) for i from 0 to n - 1 input arr[i] procedure printArr(n, arr) for i from 0 to n - 1 output arr[i], " " output newline procedure reverseArr(n, arr) for i from 0 to n / 2 - 1 temp = arr[i]

	arr[i] = arr[n - i - 1] arr[n - i - 1] = temp procedure main() output "Enter number of elements of the array: " input n declare arr[n] output "Enter elements: " readArr(n, arr) output newline output "Original array:" printArr(n, arr) output "Reversed array: " reverseArr(n, arr) printArr(n, arr) return 0
PROGRAM:	#include<stdio.h> #include<string.h> void read_array(int*a,int n) { for(int i=0;i<n;i++) scanf("%d",&a[i]); } void print_array(int*a,int n) { for(int i=0;i<n;i++) { printf("%d",a[i]); } } void reverse_arr(int*a,int n) { int b; for(int i=0;i<n/2;i++) { b=*(a+i); *(a+i)=*(a+n-1-i);

```
        *(a+n-1-i)=b;
        }
}
int main()
{
        int n;
        printf("Enter the number of elements : ");
        scanf("%d",&n);

        int a[n];
        read_array(a,n);
        printf("The array before Reverse : ");
        print_array(a,n);
        reverse_arr(a,n);
                printf("\nThe array after Reverse : ");
        print_array(a,n);
return 0;
}
```

RESULT:

```
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$ gcc largest_smallest_pointer.c
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$ ./a.out
Enter the number of elements : 5
45
78
32
64
22
The array before Reverse : 4578326422
The array after Reverse : 2264327845
The minimum = 22
 The maximum =78
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$
```

Program 3	
PROBLEM STATEMENT:	Write a program to perform matrix addition using pointers. Dimensions of matrices will be decided by the user
PSEUDO CODE:	procedure readMat(r, c, mat) for i from 0 to r - 1 for j from 0 to c - 1 input mat[i][j]

```
procedure printMat(r, c, mat)
    for i from 0 to r - 1
        for j from 0 to c - 1
            output mat[i][j], " "
        output newline

procedure addMat(r, c, mat1, mat2)
    declare mat3[r][c]
    for i from 0 to r - 1
        for j from 0 to c - 1
            mat3[i][j] = mat1[i][j] + mat2[i][j]
    output newline
    output "Addition of the matrices is:"
    printMat(r, c, mat3)

procedure main()
    output "Enter number of rows and columns of matrix 1: "
    input r1, c1
    output "Enter elements of matrix 1:"
    declare mat1[r1][c1]
    readMat(r1, c1, mat1)

    output "Enter number of rows and columns of matrix 2: "
    input r2, c2
    output "Enter elements of matrix 2:"
    declare mat2[r2][c2]
    readMat(r2, c2, mat2)

    output newline
    output "Matrix 1:"
    printMat(r1, c1, mat1)
    output "Matrix 2:"
    printMat(r2, c2, mat2)

    if r1 = r2 and c1 = c2
        addMat(r1, c1, mat1, mat2)
    else
        output "Cannot add the matrices since the matrices are not of the same
order."
```

	return 0
PROGRAM:	#include <stdio.h> void read(int r,int c,int (*p)[c]) { for(int i=0;i<r;i++) for(int j=0;j<c;j++) scanf("%d",(*(p+i)+j)); } void print(int r,int c,int (*p)[c]) { for(int i=0;i<r;i++) { for(int j=0;j<c;j++) printf("%d ",*(*(p+i)+j)); printf("\n"); } } void sum(int r,int c,int (*p1)[c],int (*p2)[c]) { for(int i=0;i<r;i++) { for(int j=0;j<c;j++) printf("%d ",*(*(p1+i)+j)+*(*(p2+i)+j)); printf("\n"); } } int main() { int r,c; printf("Enter number of rows and columns:"); scanf("%d%d",&r,&c); int m1[r][c],m2[r][c]; printf("Enter First Matrix:"); read(r,c,m1); printf("Enter Second Matrix:"); read(r,c,m2); printf("First Matrix is:\n"); print(r,c,m1);

```
        printf("Second Matrix is:\n");
        print(r,c,m2);
        printf("Sum of Two array:\n");
        sum(r,c,m1,m2);
        return 0;
}
```

RESULT:

```
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$ gcc pointer2D.c
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$ ./a.out
Enter number of rows and columns:3   2
Enter First Matrix:1
2
3
4
5
6
Enter Second Matrix:9
8
7
6
5
4
First Matrix is:
1 2
3 4
5 6
Second Matrix is:
9 8
7 6
5 4
Sum of Two array:
10 10
10 10
10 10
psipl@psipl-OptiPlex-3000:~/Desktop/2023300065$
```

CONCLUSION:	We have learnt the use of pointers to solve various problems like the addition, subtraction, finding largest and smallest numbers and also to reverse the elements in an array.