Name	Vineet Parmar
UID no.	2021300092
Experiment No.	9

AIM:	Demonstrate the use of pointers to solve a given problem.
Program 1	
PROBLEM STATEMENT :	Write a program to swap smallest and largest element in an array using pointers
ALGORITHM:	1. START 2. Define void function swap with a integer n and integer array arr[n] as parameter 3. I=0, min_index=0, max_index = 0 4. Loop from i=0 to n-1 A. If *(arr+i) < *(arr+min index) min index = i B. If *(arr + i) > *(arr + max_index) max_index = i 5. Temp = *(arr + max_index) 6. *(arr + max_index) = *(arr + min_index) 7. *(arr + min_index) = temp 8. Define main function 9. Input number of elements n 10. Input array arr[n] 11. Call function swap(n, arr) 12. Print array arr[n] 13. STOP

PROGRAM:	#include<stdio.h>
	void swap(int n, int arr[n])
	{
	int i,min_index = 0, max_index = 0;
	for(i=1;i<n;i++)
	{
	if(*(arr+i)<*(arr+min_index))
	min_index=i;

```
        if(*(arr+i)>*(arr+max_index))
            max_index=i;
    }
    int temp = *(arr + max_index);
    *(arr + max_index) = *(arr + min_index);
    *(arr + min_index) = temp;
}
int main()
{
    int n;
    printf("Enter number of elements of array: ");
    scanf("%d",&n);
    printf("Enter elements of the array: ");
    int arr[n];
    for(int i=0;i<n;i++)
        scanf("%d",(arr+i));
    swap(n,arr);
    printf("The array after swapping Maximum and Minimum elements
is:\n");
    for(int i=0;i<n;i++)
        printf("%d ",*(arr+i));
    return 0;
}
```

RESULT:

```
D:\Studies\Programs\Assignment 9>gcc -o hello As9_1.c

D:\Studies\Programs\Assignment 9>hello
Enter number of elements of array: 5
Enter elements of the array: 1
2
3
4
5
The array after swapping Maximum and Minimum elements is:
5 2 3 4 1
D:\Studies\Programs\Assignment 9>
```

Program 2	
PROBLEM STATEMENT :	Write a program to reverse the position of all elements in the array using pointers.

ALGORITHM:	1. START 2. Define void function reverse with integer n and integer array arr[n] as parameters. 3. Loop from I = 0 to n/2-1 A. temp = *(arr + i) B. *(arr + i) = *(arr + n-1 –i) C. *(arr + n-1 – i) = temp 4. Define main function 5. Input no of elements of array n 6. Input array arr[n] 7. Call function reverse(n, arr) 8. Print arr[n] 9. STOP

PROGRAM:	#include<stdio.h>
	void reverse(int n, int arr[n])
	{
	for(int i=0;i<n/2;i++)
	{
	int temp = *(arr + i);
	*(arr + i) = *(arr + n-1 - i);
	*(arr + n-1 - i) = temp;
	}
	}
	int main()
	{
	int n;
	printf("Enter number of elements of array: ");
	scanf("%d",&n);
	printf("Enter elements of the array: ");
	int arr[n];
	for(int i=0;i<n;i++)
	scanf("%d",(arr+i));
	reverse(n,arr);
	printf("The array after reversing is:\n");
	for(int i=0;i<n;i++)
	printf("%d ",*(arr+i));
	return 0;
	}

RESULT:

```
D:\Studies\Programs\Assignment 9>gcc -o hello As9_2.c

D:\Studies\Programs\Assignment 9>hello
Enter number of elements of array: 5
Enter elements of the array: 1
4
2
3
5
The array after reversing is:
5 3 2 4 1
D:\Studies\Programs\Assignment 9>
```

Program 3	
PROBLEM STATEMENT:	Write a program to perfor m matrix multiplication using pointers. Dimensions of matrices will be decided by the user.
ALGORITHM:	1. START 2. Define void function multiply with 4 integers m, n, a, b and three 2-D integer arrays arr1[m][n], arr2[a][b], arr3[m][b] 3. If n is equal to a A. Loop from I = 0 to m-1 I. Loop from J = 0 to b-1 a. sum = 0 b. Loop from k = 1 to n-1 sum += (*(*(arr1 + i) + k)) * (*(*(arr2 + k) + j)) 4. Define main function 5. Input dimensions of first matrix m and n 6. Input first matrix arr1[m][n] 7. Input dimensions of second matrix a and b 8. Input second matrix arr2[a][b] 9. Define arr3[m][b] 10. Call function multiply(m, n, a, b, arr1, arr2, arr3) 11. Print 2-D array arr3 12. STOP

PROGRAM:	#include<stdio.h>
	void multiply(int m,int n ,int arr1[m][n],int a, int b, int arr2[a][b],int arr3[m][b])
	{
	if(n==a)
	for(int i=0;i<m;i++)
	for(int j=0;j<b;j++)

```
        {
            int sum = 0;
            for(int k=0;k<n;k++)
                sum += (*(*(arr1 + i) + k)) * (*(*(arr2 + k) + j));
            *(*(arr3 + i) + j) = sum;
        }
}
int main()
{
    int m,n,a,b;
    printf("Enter dimensions of matrix ");
    scanf("%d %d",&m,&n);
    printf("Enter elements of the array: ");
    int arr1[m][n];
    for(int i=0;i<m;i++)
        for(int j=0;j<n;j++)
            scanf("%d",(*(arr1 + i) + j));
    printf("Enter dimensions of matrix ");
    scanf("%d %d",&a,&b);
    printf("Enter elements of the array: ");
    int arr2[a][b];
    for(int i=0;i<a;i++)
        for(int j=0;j<b;j++)
            scanf("%d",(*(arr2 + i) + j));
    int arr3[m][b];
    multiply(m,n,arr1,a,b,arr2,arr3);
    printf("The matrix after multiplying is:\n");
    for(int i=0;i<m;i++)
        for(int j=0;j<b;j++)
            printf("%d ", *(*(arr3 + i) + j));
    return 0;
}
```

RESULT:

```
D:\Studies\Programs\Assignment 9>gcc -o hello As9_3.c

D:\Studies\Programs\Assignment 9>hello
Enter dimensions of matrix 2 2
Enter elements of the array: 1
2
3
4
Enter dimensions of matrix 2 2
Enter elements of the array: 1
2
3
4
The matrix after multiplying is:
7 10 15 22
D:\Studies\Programs\Assignment 9>
```

CONCLUSION:	In this experiment, we learned how to take inputs, print outputs of 1D arrays and how to perfom operations on them using pointers.