PRACTICAL SET 2

Aim: Implementation of Array, String and pointer concepts.

Array is a collection of homogeneous elements, i.e. elements belonging to same data type. String is a sequence of characters, more specifically, it can be called a character array. Pointer stores address of another variable. All three make up the derived data types in C language. This practical helps the student to understand logical manipulation of these types through implementation.

Requirements:

1) A desktop computer system

2) Code Blocks IDE

PRACTICAL-1

Aim-WAP to take 10 values from the user and store them in an array and Print the elements stored in the array.

Input:-

```
#include <stdio.h>

int main()
{
   int arr[10];
   int i;
      printf("\n\nRead and Print elements of an array:\n");
      printf("                                        \n");
   printf("Input 10 elements in the array :\n");
   for(i=0; i<10; i++)
   {
           printf("element - %d : ",i);
    scanf("%d", &arr[i]);
   }
   printf("\nElements in array are: ");
   for(i=0; i<10; i++)
   {
    printf("%d  ", arr[i]);
   }
   printf("\n");
   return 0;
}
```

Output:-

```
Read and Print elements of an array:

Input 10 elements in the array :
element - 0 : 1
element - 1 : 2
element - 2 : 3
element - 3 : 4
element - 4 : 5
element - 5 : 6
element - 6 : 7
element - 7 : 8
element - 8 : 9
element - 9 : 0

Elements in array are: 1  2  3  4  5  6  7  8  9  0
```

PRACTICAL-2

Aim-WAP to find out the average of n numbers using arrays

Input:-

```
#include <stdio.h>
int main() {
   int n, i;
   float num[100], sum = 0.0, avg;
   printf("Enter the numbers of elements: ");
   scanf("%d", &n);
   while (n > 100 || n < 1) {
      printf("Error! number should in range of (1 to 100).\n");
      printf("Enter the number again: ");
      scanf("%d", &n);
   }
   for (i = 0; i < n; ++i) {
      printf("%d. Enter number: ", i + 1);
      scanf("%f", &num[i]);
      sum += num[i];
   }
   avg = sum / n;
   printf("Average = %.2f", avg);
   return 0;
}
```

Output:-

```
1. Enter number: 21
2. Enter number: 23
3. Enter number: 43
4. Enter number: 11
5. Enter number: 20
Average = 23.60
```

PRACTICAL-3

Aim-WAP to get positive integers from a user into an array of size 10 and calculate the number of odd and even integers in the array.

Input:-

```
#include<stdio.h>
int main()
{
 int Size, i, a[10];
 int Even_Count = 0, Odd_Count = 0;
 printf("\n Please Enter the Size of an Array :  ");
 scanf("%d", &Size);
 printf("\nPlease Enter the Array Elements\n");
 for(i = 0; i < Size; i++)
 {
   scanf("%d", &a[i]);
 }
 for(i = 0; i < Size; i ++)
{
   if(a[i] % 2 == 0)
   {
     Even_Count++;
   }
   else
   {
     Odd_Count++;
   }
 }
 printf("\n Total Number of Even Numbers in this Array = %d ", Even_Count);
 printf("\n Total Number of Odd Numbers in this Array = %d ", Odd_Count);
 return 0;
}
```

Output:-

```
 Please Enter the Size of an Array :  5
Please Enter the Array Elements
11
33
44
66
77
 Total Number of Even Numbers in this Array = 2
 Total Number of Odd Numbers in this Array = 3
```

PRACTICAL-4

Aim-Write a program in C to copy the elements of one array into another array

Input:-

```
#include <stdio.h>
int  main()
{
   int arr1[100], arr2[100];
   int i, n;
     printf("\n\nCopy the elements one array into another array :\n");
     printf("--------------------------------------------------\n");
     printf("Input the number of elements to be stored in the array :");
     scanf("%d",&n);
     printf("Input %d elements in the array :\n",n);
     for(i=0;i<n;i++)
      {
            printf("element - %d : ",i);
            scanf("%d",&arr1[i]);
           }
   for(i=0; i<n; i++)
   {
      arr2[i] = arr1[i];
   }
   printf("\nThe elements stored in the first array are :\n");
   for(i=0; i<n; i++)
   {
      printf("% 5d", arr1[i]);
   }
  printf("\n\nThe elements copied into the second array are :\n");
   for(i=0; i<n; i++)
   {
      printf("% 5d", arr2[i]);
   }
             printf("\n\n");
return 0;
}
```

Output:-

```
Copy the elements one array into another array :
--------------------------------------------------
Input the number of elements to be stored in the array :5
Input 5 elements in the array :
element - 0 : 1
element - 1 : 2
element - 2 : 3
element - 3 : 4
element - 4 : 5
The elements stored in the first array are :
    1    2    3    4    5
The elements copied into the second array are :
    1    2    3    4    5
```

PRACTICAL-5

Aim-(a) WAP to insert an element from the given array.

(b) WAP to delete an element from the given array.

(B):-

Input:-

```
#include <stdio.h>
int main()
{
  int array[100], position, c, n;
  printf("Enter number of elements in array\n");
  scanf("%d", &n);
  printf("Enter %d elements\n", n);
  for (c = 0; c < n; c++)
    scanf("%d", &array[c]);
  printf("Enter the location where you wish to delete element\n");
  scanf("%d", &position);
  if (position >= n+1)
    printf("Deletion not possible.\n");
  else
  {
    for (c = position - 1; c < n - 1; c++)
      array[c] = array[c+1];
    printf("Resultant array:\n");
    for (c = 0; c < n - 1; c++)
      printf("%d\n", array[c]);
  }
  return 0;
}
```

Output:-

```
Enter number of elements in array
5
Enter 5 elements
1
2
3
4
5
Enter the location where you wish to delete element
3
Resultant array:
1
2
4
5
```

(A)

Input:-

```
#include <stdio.h>

int main()
{
  int array[100], position, c, n, value;
  printf("Enter number of elements in array\n");
  scanf("%d", &n);
  printf("Enter %d elements\n", n);
  for (c = 0; c < n; c++)
    scanf("%d", &array[c]);
  printf("Enter the location where you wish to insert an element\n");
  scanf("%d", &position);
  printf("Enter the value to insert\n");
  scanf("%d", &value);
  for (c = n - 1; c >= position - 1; c--)
    array[c+1] = array[c];
  array[position-1] = value;
  printf("Resultant array is\n");
  for (c = 0; c <= n; c++)
    printf("%d\n", array[c]);
  return 0;
}
```

Output:-

```
Enter number of elements in array
5
Enter 5 elements
1
2
3
4
5
Enter the location where you wish to insert an element
3
Enter the value to insert
6
Resultant array is
1
2
6
3
4
5
```

PRACTICAL-6

Aim-WAP to search an element and its position from the given array.

Input:-

```
#include <stdio.h>
int main()
{
   int a[10000],i,n,search;
   printf("Enter size of the  array : ");
   scanf("%d", &n);
   printf("Enter elements in array : ");
   for(i=0; i<n; i++)
   {
      scanf("%d",&a[i]);
   }
    printf(" search element  : ");
   scanf("%d", &search);
   for(i=0; i<n; i++)
   {
      if(a[i]==search)
      {
                         printf("element found ");
         return 0;
      }

   }
        printf("element  not  found");
 return 0;
}
```

Output:-

```
Enter size of the  array : 5
Enter elements in array : 1
2
3
4
5
 search element  : 6
element  not
```

PRACTICAL-7

Aim-WAP to merging an element from the given two array.

Input:-

```
#include<stdio.h>
int main()
{
   int arr1[50], arr2[50], a,b, i, k, merge[100];
   printf("Enter Array 1 Size: ");
   scanf("%d", &a);
   printf("Enter Array 1 Elements: ");
   for(i=0; i<a; i++)
   {
      scanf("%d", &arr1[i]);
      merge[i] = arr1[i];
   }
   k = i;
   printf("\nEnter Array 2 Size: ");
   scanf("%d", &b);
   printf("Enter Array 2 Elements: ");
   for(i=0; i<b; i++)
   {
      scanf("%d", &arr2[i]);
      merge[k] = arr2[i];
      k++;
   }
   printf("\nThe new array after merging is:\n");
   for(i=0; i<k; i++)
      printf("%d ", merge[i]);
   return 0;
}
```

Output:-

```
Enter Array 1 Size: 4
Enter Array 1 Elements: 1
2
3
4
Enter Array 2 Size: 7
Enter Array 2 Elements: 12
23
45
33
55
76
8
The new array after merging is:
1 2 3 4 12 23 45 33 55 76 8
```

PRACTICAL-8

Aim-WAP to find the minimum(smallest) and maximum(largest) value and its position in array from the given numbers using array.

Input:-

```
#include<stdio.h>
int main()
{
   int a[100], n, i, max, min, maxPos, minPos;
   printf("Enter array size : ");
   scanf("%d",&n);
   maxPos=minPos=0;
  printf("Enter array elements: ");
   scanf("%d",&a[0]);
   max=min=a[0];
   for(i=1; i<n; i++)
   {
      scanf("%d",&a[i]);
      if(max<a[i])
      {
         min=a[i];
         minPos=i;
      }
      if(min>a[i])
      {
         min=a[i];
         minPos=i;
      }
   }
   printf("Largest element is %d at %d position.\n", max,maxPos);
   printf("Smallest element is %d at %d position.", min,minPos);
   return 0;
}
```

Output:-

```
Enter array size : 5
Enter array elements: 1
2
3
4
5
Largest element is 1 at 0 position.
Smallest element is 5 at 4 position.
```

PRACTICAL-9

Aim-

(a)WAP to find first 2 maximum number form the given value or number using array

(b)WAP to find first 2 minimum number form the given value or number using array.

(A)

Input:-

```
#include <stdio.h>
int main ()
{
   int n = 0, i = 0, maximum1 = 0, maximum2 = 0, temp = 0;
   printf("Enter the size of the array :",n);
   scanf("%d", &n);
   int array[n];
   printf("Enter the elements\n");
   for (i = 0; i < n; i++)
   {
      scanf ("%d", &array[i]);
   }
   printf("The array elements are : \n");
   for (i = 0; i < n; i++)
   {
      printf ("%d\t", array[i]);
   }
   printf ("\n");
   maximum1 = array[0];
   maximum2 = array[1];
   if (maximum1 < maximum2)
   {
      temp = maximum1;
      maximum1 = maximum2;
      maximum2 = temp;
   }
   for (int i = 2; i < n; i++)
   {
      if (array[i] > maximum1)
      {
         maximum2 = maximum1;
         maximum1 = array[i];
      }
      else if (array[i] > maximum2 && array[i] != maximum1)
      {
         maximum2 = array[i];
      }
   }
   printf ("The FIRST LARGEST = %d\n", maximum1);
   printf ("THE SECOND LARGEST = %d\n", maximum2);
   return 0;
}
```

Output:-

```
Enter the size of the array :5
Enter the elements
```

```
1
2
3
4
5
The array elements are :
1	2	3	4	5
The FIRST MAXIMUM = 5
THE SECOND MAXIMUM = 4
```

(B)

Input:-

```
#include <stdio.h>
int main ()
{
   int n = 0, i = 0, minimum1 = 0, minimum2 = 0, temp = 0;
   printf("Enter the size of the array :");
   scanf("%d", &n);
   int array[n];
   printf("Enter the elements\n");
   for (i = 0; i < n; i++)
   {
      scanf ("%d", &array[i]);
   }
   printf("The array elements are : \n");
   for (i = 0; i < n; i++)
   {
      printf ("%d\t", array[i]);
   }
   printf ("\n");
   minimum1 = array[0];
   minimum2 = array[1];
   if (minimum1 > minimum2)
   {
      temp = minimum1;
      minimum1 = minimum2;
      minimum2 = temp;
   }
   for (int i = 2; i < n; i++)
   {
      if (array[i] < minimum1)
      {
         minimum2 = minimum1;
         minimum1 = array[i];
      }
      else if (array[i] < minimum2 && array[i] !=minimum1)
      {
         minimum2 = array[i];
      }
   }
   printf ("The FIRST MINIMUM = %d\n", minimum1);
```

```
    printf ("THE SECOND MINIMUM = %d\n", minimum2);
    return 0;
}
```

Output:-

```
Enter the size of the array :5
Enter the elements
1
2
3
4
5
The array elements are :
1       2       3       4       5
The FIRST MINIMUM = 1
THE SECOND MINIMUM = 2
```

PRACTICAL-10

Aim-

(a)WAP to find Sum of N Input Numbers using Array

(b)WAP to get a set of positive integers from user and store it in an array of size 10. Calculate the sum of odd integers and product of even integers.

(A)

Input:-

```
#include <stdio.h>
int main()
{
   int n, sum = 0, c;
   printf("enter the array :");
   scanf("%d", &n);
   int array[n];

   for (c = 0; c < n; c++)
   {
      scanf("%d", &array[c]);
      sum = sum + array[c];
   }

   printf("Sum = %d\n", sum);

   return 0;
}
```

Output:-

```
enter the array :5
1
2
3
4
5
Sum = 15
```

(B)

Input:-

```
#include<stdio.h>

int main()
{
    int n,c,even=0,odd=0;
    printf("enter the array :");
    scanf("%d",&n);
    int array[n];
    for (c=1; c<=n;  c++)
    {
        scanf("%d",&array[n]);

      if (c %2 == 0 )
      {
          even +=array[n];
      }
      else
      {
          odd +=array[n];
      }
    }
       printf("sum of odd numbers: %d\n",odd);
       printf("sum of even numbers: %d\n",even);
   return 0;
}
```

Output:-

```
enter the array :5
1
2
3
4
5
sum of odd numbers: 9
sum of even numbers: 6
```

PRACTICAL-11

Aim-(a)WAP to sort an array Ascending order
(b)WAP to sort an array Descending order

(A)

Input:-

```
 #include<stdio.h>
 int main()
   {
      int i, j, a, n;
      printf("Enter the value of array: \n");
      scanf("%d", &n);
      int array[n];
      printf("Enter the numbers \n");
      for (i = 0; i < n; ++i)
         scanf("%d", &array[i]);
      for (i = 0; i < n; ++i)
      {
         for (j = i + 1; j < n; ++j)
         {
            if (array[i] >array[j])
            {
               a =  array[i];
               array[i] = array[j];
               array[j] = a;
            }
         }
      }
      printf("The numbers arranged in ascending order are given below \n");
      for (i = 0; i < n; ++i)
         printf("%d\n", array[i]);
return 0;
   }
```

Output:-

```
Enter the value of array:
5
Enter the numbers
34
56
23
12
78
The numbers arranged in ascending order are given below
12
23
34
56
78
```

(B)

Input:-

```
 #include<stdio.h>
 int main()
 {
     int i, j, a, n;
     printf("Enter the value of array \n");
     scanf("%d", &n);
     int array[n];
     printf("Enter the numbers \n");
     for (i = 0; i < n; ++i)
        scanf("%d", &array[i]);
     for (i = 0; i < n; ++i)
     {
        for (j = i + 1; j < n; ++j)
        {
           if (array[i] <array[j])
           {
              a =  array[i];
              array[i] = array[j];
              array[j] = a;
           }
        }
     }
     printf("The numbers arranged in ascending order are given below \n");
     for (i = 0; i < n; ++i)
        printf("%d\n", array[i]);
return 0;
 }
```

Output:-

```
Enter the value of array
5
Enter the numbers
44
55
67
32
45
The numbers arranged in ascending order are given below
67
55
45
44
32
```

PRACTICAL-12

Aim-WAP to print a given array in reverse order

Input:-

```
#include<stdio.h>

int main ()
{
   int n,c,a;
   printf("enter the value of array:");
   scanf("%d",&n);
   int array[n];
   for (c=0;c<n;c++)
   {
      scanf("%d",&array[c]);
   }
   printf("reverse order : \n");
   for (c=n-1;c>=0;c--)
      {
         printf("%d \n",array[c]);
      }
   return 0;
}
```

Output:-

```
enter the value of array:5
1
2
3
4
5
reverse order :
5
4
3
2
1
```

PRACTICAL-13

Aim-WAP to delete the repeated value from the elements of an array.

Input:-

```
#include<stdio.h>
int  main()
{
   int  a, b, c, n;
   printf("\nEnter array size: ");
   scanf("%d", &n);
   int arr[n];
   printf("\nEnter %d array element: ", n);
   for(a = 0; a < n; a++)
   {
      scanf("%d", &arr[a]);
   }
   printf("\nOriginal array is: ");
   for(a = 0; a < n; a++)
   {
      printf(" %d", arr[a]);
   printf("\nNew array is: ");
   for(a = 0; a < n; a++)
   {
      for(b = a+1; b < n; )
      {
         if(arr[b] == arr[a])
         {
            for(c = b; c < n; c++)
            {
               arr[c] = arr[c+1];
            }
            n--;
         }
         else
         {
            b++;
         }
      }
   }
   for(a = 0; a < n; a++)
   {
      printf("%d ", arr[a]);
   }
   }
  return 0 ;
}
```

Output:-

```
Enter array size: 6

Enter 6 array element: 1
2
3
```

```
4
5
5

Original array is:  1
New array is: 1 2 3 4 5
```

PRACTICAL-14

Aim-WAP to read and print a Row and Column Matrix, where value of Row and Column must be entering by User.

Input:-

```
#include <stdio.h>

int main()
{
    int z, x, c, v,a,b;
    int arr[10][20];
    printf("Enter number of rows : ");
    scanf("%d", &c);
    printf("Enter number of columns : ");
    scanf("%d", &v);
    /* Input data in matrix */
    for (z = 0; z < c; z++)
    {
        for (x = 0; x < v; x++)
        {
            printf("Enter data in [%d][%d]: ", z, x);
            scanf("%d", &arr[z][x]);
        }
    }
    /* Display the matrix */
    for (z = 0; z < c; z++)
    {
        for (x = 0; x < v; x++)
        {
            printf("%d\t", arr[z][x]);
        }
        printf("\n");
    }
    return 0;
}
```

Output:-

```
Enter number of rows : 2
Enter number of columns : 2
Enter data in [0][0]: 1
Enter data in [0][1]: 23
Enter data in [1][0]: 54
Enter data in [1][1]: 7
1       23
54      7
```

PRACTICAL-15

Aim-WAP to read a matrix and find the addition and multiplication of all elements of two dimensional matrix array.

Input:-

```
#include <stdio.h>
int main() {
   int r, c, a[100][100], b[100][100], sum[100][100], i, j,mult[100][100];
   printf("Enter the number of rows (between 1 and 100): ");
   scanf("%d", &r);
   printf("Enter the number of columns (between 1 and 100): ");
   scanf("%d", &c);
//first matrix
   printf("\nEnter elements of 1st matrix:\n");
   for (i = 0; i < r; ++i)
      for (j = 0; j < c; ++j) {
         printf("Enter element a%d%d: ", i + 1, j + 1);
         scanf("%d", &a[i][j]);
      }
   for (i = 0; i < r; i++)
   {
      for (j = 0; j < c; j++)
      {
         printf("%d\t", a[i][j]);
      }
      printf("\n");
   }
//second matrix
   printf("Enter elements of 2nd matrix:\n");
   for (i = 0; i < r; ++i)
      for (j = 0; j < c; ++j) {
         printf("Enter element a%d%d: ", i + 1, j + 1);
         scanf("%d", &b[i][j]);
      }
   for (i = 0; i < r; i++)
   {
      for (j = 0; j < c; j++)
      {
         printf("%d\t", b[i][j]);
      }
      printf("\n");
   }
   // adding two matrices
   for (i = 0; i < r; ++i)
      for (j = 0; j < c; ++j) {
         sum[i][j] = a[i][j] + b[i][j];
      }
   // printing the result
   printf("\nSum of two matrices: \n");
   for (i = 0; i < r; ++i)
      for (j = 0; j < c; ++j) {
         printf("%d   ", sum[i][j]);
```

```
            if (j == c - 1) {
                printf("\n\n");
            }
        }
        // mult. two matrices
    for ( i = 0; i < r; i++)
    {
        for( j = 0; j < c; j++)
        {
            mult[i][j]=a[i][j]*b[i][j];
        }
    }
    //printing the result
    printf("\n multiplication of both matrix: \n");
    for ( i = 0; i < r; i++)
    {
        for ( j = 0; j < c; j++)
        {
           printf("%d\t",mult[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Output:-

```
Enter the number of rows (between 1 and 100): 2
Enter the number of columns (between 1 and 100): 2

Enter elements of 1st matrix:
Enter element a11: 12
Enter element a12: 34
Enter element a21: 54
Enter element a22: 67
12      34
54      67
Enter elements of 2nd matrix:
Enter element a11: 12
Enter element a12: 34
Enter element a21: 32
Enter element a22: 21
12      34
32      21

Sum of two matrices:
24   68

86   88


 multiplication of both matrix:
144     1156
1728    1407
```

PRACTICAL-16

Aim-WAP to transpose of a matrix

Input:-

```
#include <stdio.h>
int main() {
   int a[10][10], t[10][10], r, c, i, j;
   printf("Enter rows and columns: ");
   scanf("%d %d", &r, &c);
   // Assigning elements to the matrix
   printf("\nEnter matrix elements:\n");
   for (i = 0; i < r; ++i)
      for (j = 0; j < c; ++j) {
         printf("Enter element a%d%d: ", i + 1, j + 1);
         scanf("%d", &a[i][j]);
      }
   // Displaying the matrix a[][]
   printf("\nEntered matrix: \n");
   for (i = 0; i < r; ++i)
      for (j = 0; j < c; ++j) {
         printf("%d  ", a[i][j]);
         if (j == c - 1)
            printf("\n");
      }
       for (i = 0; i < r; i++)
   {
      for (j = 0; j < c; j++)
      {
         printf("%d\t", a[i][j]);
      }
      printf("\n");
   }
   // Finding the transpose of matrix a
   for (i = 0; i < r; ++i)
      for (j = 0; j < c; ++j) {
         t[j][i] = a[i][j];
      }
   // Displaying the transpose of matrix a
   printf("\nTranspose of the matrix:\n");
   for (i = 0; i < c; ++i)
      for (j = 0; j < r; ++j) {
         printf("%d  ", t[i][j]);
         if (j == r - 1)
            printf("\n");
      }
   return 0;
}
```

Output:-

```
Enter rows and columns: 2
2
```

```
Enter matrix elements:
Enter element a11: 21
Enter element a12: 34
Enter element a21: 32
Enter element a22: 12

Entered matrix:
21  34
32  12
21      34
32      12

Transpose of the matrix:
21  32
34  12
```

PRACTICAL-17

Aim-a.WAP to find addition of values of two 3 X 3 matrices.

b.WAP to find multiplication of two 3 X 3 matrices.

Input:-

Both in one program :—

```
#include <stdio.h>
int main() {
    int r, c, a[100][100], b[100][100], sum[100][100], i, j,mult[100][100];
    printf("Enter the number of rows (between 1 and 100): ");
    scanf("%d", &r);
    printf("Enter the number of columns (between 1 and 100): ");
    scanf("%d", &c);
//first matrix
    printf("\nEnter elements of 1st matrix:\n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            printf("Enter element a%d%d: ", i + 1, j + 1);
            scanf("%d", &a[i][j]);
        }
    for (i = 0; i < r; i++)
    {
        for (j = 0; j < c; j++)
        {
            printf("%d\t", a[i][j]);
        }
        printf("\n");
    }
//second matrix
    printf("Enter elements of 2nd matrix:\n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            printf("Enter element a%d%d: ", i + 1, j + 1);
            scanf("%d", &b[i][j]);
        }
    for (i = 0; i < r; i++)
    {
        for (j = 0; j < c; j++)
        {
            printf("%d\t", b[i][j]);
        }
        printf("\n");
    }
    // adding two matrices
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            sum[i][j] = a[i][j] + b[i][j];
        }
    // printing the result
    printf("\nSum of two matrices: \n");
    for (i = 0; i < r; ++i)
```

```
        for (j = 0; j < c; ++j) {
            printf("%d   ", sum[i][j]);
            if (j == c - 1) {
                printf("\n\n");
            }
        }
        // mult. two matrices
    for ( i = 0; i < r; i++)
    {
        for( j = 0; j < c; j++)
        {
            mult[i][j]=a[i][j]*b[i][j];
        }
    }
    //printing the result
    printf("\n multiplication of both matrix: \n");
    for ( i = 0; i < r; i++)
    {
        for ( j = 0; j < c; j++)
        {
           printf("%d\t",mult[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

Output:-

```
Enter the number of rows (between 1 and 100): 3
Enter the number of columns (between 1 and 100): 3

Enter elements of 1st matrix:
Enter element a11: 21
Enter element a12: 34
Enter element a13: 56
Enter element a21: 78
Enter element a22: 32
Enter element a23: 12
Enter element a31: 22
Enter element a32: 33
Enter element a33: 55
21      34      56
78      32      12
22      33      55
Enter elements of 2nd matrix:
Enter element a11: 1
Enter element a12: 2
Enter element a13: 3
Enter element a21: 4
Enter element a22: 5
Enter element a23: 6
Enter element a31: 7
Enter element a32: 8
Enter element a33: 9
1       2       3
4       5       6
7       8       9

Sum of two matrices:
22   36   59

82   37   18

29   41   64


 multiplication of both matrix:
21      68      168
312      160     72
154      264     495
```

PRACTICAL-18

Aim-a.WAP in C to input a string and print it.

b.WAP in C to find the length of a string without using library function.

c.WAP to find the length of a String with using any standard library function

Input:-

a)

```
#include<stdio.h>
//Write a program in C to input a string and print it.
int main()
{
   char a[100];
   printf("enter a string :");
   scanf("%s",a);
   printf("your string : %s\n",a);
   return 0;
}
```

B)

```
#include<stdio.h>
#include<string.h>
//WAP in C to find the length of a string without using library function.
int main()
{
   char s[100];
   int a,b=0;
   printf("\nfind the leangth of a string:\n");
   printf("\n");
   printf("enter the string: ");
   scanf("%s",s);
   for ( a = 0; s[a]!='\0' ; a++)
   {
      b++;
   }
   printf("the string %d number of characters.\n",b);
   printf("the length of the string %s is:%d\n",s,b);
   return 0;
}
```

C)

```
#include<stdio.h>
#include<string.h>
//WAP to find the length of a String with using any standard library function
int main ()
{
   char s[100];
  int a;

  printf("\nEnter the String : ");
  gets(s);

  a = strlen(s);

  printf("\nLength of Given String : %d\n", a);
```

```
   return 0;
}
```

Output:-

A)

```
enter a string :hello
your string : hello
```

B)

```
find the leangth of a string:

enter the string: hello
the string 5 number of characters.
the length of the string hello is:5
```

C)

```
warning: this program uses gets(), which is unsafe.
Enter the String : hello

Length of Given String : 5
```

PRACTICAL-19

Aim-a.Write a program in C to copy one string to another string with using strcpy function.

b.Write a program in C to copy one string to another string without using strcpy function.

Input:-

A)

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

int main()
{
   char a[100],b[100];
   printf("\n Enter the string a:");
   gets(a);
   strcpy(b,a);
   printf("\n copy string :%s \n",b);
   return 0;
}
```

B)

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

int main()
{
   char a[100],b[100],i;
   printf("enter string a: ");
   fgets(a,sizeof(a),stdin);
   for (i=0;a[i]!= '\0';i++)
   {
      b[i]=a[i];
   }
   b[i]= '\0';
   printf("string b:%s",b);
   return 0;
}
```

Output:-

A)

```
warning: this program uses gets(), which is unsafe.
 Enter the string a:hello operator

 copy string :hello operator
```

B)

```
enter string a: hello world
string b:hello world
```

PRACTICAL-20

Aim-a.Write a program in C to compare two strings with using string library functions.
b.Write a program in C to compare two strings without using string library functions.

Input:-

A)

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

int main()
{
   char a[100],b[100];
   int c;
   printf("enter the first string a:");
   gets(a);
   printf("enter the second string b:");
   gets(b);
   c=strcmp(a,b);
   if(c==0)
   {
      printf("strings are same\n");
   }
   else
   {
      printf("strings are not same\n");
   }
   return 0;
}
```

B)

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

int main()
{
   char a[100],b[100];
   int c=0,i,j;
   printf("enter the first string a:");
   gets(a);
   printf("enter the second string b:");
   gets(b);
   i=0;
   j=0;
   while (a[i]!='\0')
   {
      i++;
   }
   while (b[j]!='\0')
   {
      j++;
   }
   if (i!=j)
   {
      c=0;
   }
   else
   {
      for ( i = 0,j=0;a[i] != '\0',b[j] != '\0'; i++,j++)
      {
          if (a[i]==b[j])
          {
              c==1;
          }

      }

   }
   if(c==0)
   {
      printf("strings are same\n");
   }
   else
   {
      printf("strings are not same\n");
   }
   return 0;
}
```

Output:-

A)

```
enter the first string a:hello
enter the second string b:hello john
strings are not same
```

B)

```
enter the first string a:hello
enter the second string b:hello
strings are same
```

PRACTICAL-21

Aim-a.Write a program in C to count the total number of words in a string.
b.Write a program in C to count total number of alphabets, digits and special characters in a string.

Input:-

A)

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

int main()
{
   char a[200];
   int c = 0, i;

   printf("Enter the string:\n");
   gets(a);
   for (i = 0;a[i] != '\0';i++)
   {
      if (a[i] == ' ' && a[i+1] != ' ')
         c++;
   }
   printf("Number of words in the string are: %d\n", c + 1);
   return 0;
}
```

B)

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

int main ()
{
   char a[100];
   int b,c,e,i;
   b=c=e=i=0;
   printf("\n\n count total number : \n");
   printf("\n");
   printf("input the string :");
   fgets(a, sizeof a,stdin);
   while (a[i]!='\0')
   {
      if ((a[i]>='a'&&a[i]<='z')||(a[i]>='A'&&a[i]<='Z'))
      {
         b++;
      }
      else if (a[i]>='0'&&a[i]<='9')
      {
         c++;
      }
      else
      {
         e++;
      }
```

```
        i++;
    }
    printf("alphabets in string:%d\n",b);
    printf("digits in string :%d\n",c);
    printf("special characters in string:%d\n",e);
return 0;
}
```

Output:-

A)

```
enter the string:
hello
Number of words in given string are: 1
```

B)

```
count total number :

input the string :hello guys welcome to 123
alphabets in string:18
digits in string :3
special characters in string:5
```

PRACTICAL-22

Aim-a.WAP to concatenate two strings with using strcat standard library function.

b.WAP to concatenate two strings without using strcat standard library function..

Input:-

A)

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

int main()
{
   char a[100], b[100];

   printf("Enter the first string:\n");
   gets(a);

   printf("Enter the second string:\n");
   gets(b);

   strcat(a,b);

   printf("String  concatenation is: %s\n",a);

   return 0;

}
```

B)

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

int main()
{
   char a[100], b[100];
   int j,l;
   printf("Enter the first string:\n");
   gets(a);
   printf("Enter the second string:\n");
   gets(b);
    l = 0;
  while (a[l] != '\0') {
    ++l;
  }
  // concatenate s2 to s1
  for (j = 0; b[j] != '\0'; ++j, ++l) {
    a[l] = b[j];
  }
  // terminating the s1 string
  a[l] = '\0';
  printf("String concatenation is: %s\n",a);
  return 0;
}
```

Output:-

A)

```
Enter the first string:
hello
Enter the second string:
guys
String concatenation is: hello guys
```

B)

```
Enter the first string:
hello
Enter the second string:
guys welcome to 123
String concatenation is: hello guys welcome to 123
hello guys welcome to 123
```

PRACTICAL-23

Aim-a.WAP to find the number of times a given word 'the' appears in the given string.

b.Write a program in C to Find the Frequency of Characters.

Input:-

A)

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

int main()
{
       int ctr=0,i,freq=0;
      int t,h,e,spc;
       char str[100];
     printf("\n\nFind the number of times the word 'the ' in any combination appears :\n");
     printf("------------------------------------------------------------------------\n");
       printf("Input the string : ");
       fgets(str,sizeof str,stdin);
      ctr=strlen(str);
      /*Counts the frequency of the word 'the' with a trailing space*/
      for(i=0;i<=ctr-3;i++)
      {
             t=(str[i]=='t'||str[i]=='T');
             h=(str[i+1]=='h'||str[i+1]=='H');
             e=(str[i+2]=='e'||str[i+2]=='E');
             spc=(str[i+3]==' '||str[i+3]=='\0');
             if ((t&&h&&e&&spc)==1)
               freq++;
      }
       printf("The frequency of the word \'the\' is : %d\n\n",freq);
   return 0;
}
```

B)

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

int main() {
   char a[1000], ch;
   int c = 0;
   printf("Enter a string: ");
   fgets(a, sizeof(a), stdin);
   printf("Enter a character : ");
   scanf("%c", &ch);
   for (int i = 0; a[i] != '\0'; ++i) {
      if (ch == a[i])
         ++c;
   }
   printf("Frequency of %c = %d \n", ch, c);
   return 0;
}
```

Output:-

A)

```
Find the number of times the word 'the ' in any combination appears :
----------------------------------------------------------------------
Input the string : find the value of the world
The frequency of the word 'the' is : 2
```

B)

```
Enter a string: hello guys
Enter a character : l
Frequency of l = 2
```

PRACTICAL-24

Aim-WAP to reverse a string given by user without using string functions.

Input:-

```
#include<stdio.h>
#include<string.h>
//Write a program in C to print individual characters of string in reverse order.

int main()
{
   char s[100]; /* Declares a string of size 100 */
   int a=0;

      printf("\n\nPrint individual characters of string in reverse order :\n");
      printf("\n");
      printf("Input the string : ");
      fgets(s, sizeof s, stdin);
          a=strlen(s);
          printf("The characters of the string in reverse are : \n");
      for(s[a]='\0';a>=0;a--)
       {
         printf("%c", s[a]);
       }
   printf("\n");
   return 0;
}
```

Output:-

```
Print individual characters of string in reverse order :

Input the string : hello
The characters of the string in reverse are :

olleh
```

PRACTICAL-25

Aim-WAP to check whether string entered by user is palindrome or not.

Input:-

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

int main()
{
   char a[1000];
   int i,n,c=0;
   printf("Enter  the string : ");
   gets(a);
   n=strlen(a);
   for(i=0;i<n/2;i++)
   {
      if(a[i]==a[n-i-1])
      c++;
      }
      if(c==i)
         printf("string is palindrome \n");
   else
      printf("string is not palindrome\n");
   return 0;
}
```

Output:-

```
Enter  the string : hello
string is not palindrome
```

PRACTICAL-26

Aim-WAP to sort a string in alphabetical order by swapping the characters in the string

Input:-

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

int main ()
{
	char a[100];
  printf("\nEnter the string : ");
	gets(a);
	char temp;
	int i, j;
	int n = strlen(a);
	for (i = 0; i < n-1; i++) {
		for (j = i+1; j < n; j++) {
			if (a[i] > a[j]) {
					temp = a[i];
					a[i] = a[j];
					a[j] = temp;
			}
		}
	}
	printf("The sorted string is : %s\n",a);
	return 0;
}
```

Output:-

```
Enter the string : face
The sorted string is : acef
```

PRACTICAL-27

Aim-WAP to read a sentence and replace lowercase characters by uppercase and vice-versa.

Input:-

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

int main()
{
  char s[100];
  int c, ch, i;
       printf("\n\nReplace lowercase characters by uppercase and vice-versa :\n");
       printf("------------------------------------------------------------\n");
           printf("Input the string : ");
       fgets(s, sizeof s, stdin);
  i=strlen(s);
  c = i; /*shows the number of chars accepted in a sentence*/
  printf("\nThe given sentence is   : %s",s);
  printf("After Case changed the string  is: ");
  for(i=0; i < c; i++)
  {
    ch = islower(s[i]) ? toupper(s[i]) : tolower(s[i]);
    putchar(ch);
   }
   printf("\n\n");
return 0;
}
```

Output:-

```
Replace lowercase characters by uppercase and vice-versa :
------------------------------------------------------------
Input the string : hello THis

The given sentence is   : hello THis
After Case changed the string  is: HELLO thIS
```

PRACTICAL-28
Aim-WAP to Swap Strings in C
Input:-

```
#include <stdio.h>
#include <string.h>
int main()
{
  char a[100], b[100], t[100];

  printf("Enter first string\n");
  gets(a);

  printf("Enter second string\n");
  gets(b);

  printf("\nBefore Swapping\n");
  printf("First string: %s\n", a);
  printf("Second string: %s\n\n", b);

  strcpy(t, a);
  strcpy(a, b);
  strcpy(b, t);

  printf("After Swapping\n");
  printf("First string: %s\n", a);
  printf("Second string: %s\n", b);

  return 0;
}
```

Output:-

```
Enter first string
hello
Enter second string
guys

Before Swapping
First string: hello
Second string: guys

After Swapping
First string: guys
Second string: hello
```

PRACTICAL-29

Aim-WAP in C to store n elements in an array and print the elements using pointer.

Input:-

```
#include <stdio.h>
int main() {
   int d[100],n;
   printf("enter the value of n:");
   scanf("%d",&n);
   printf("Enter elements: ");
   for (int i = 0; i < n; ++i)
      scanf("%d", d + i);

   printf("You entered: \n");
   for (int i = 0; i < n; ++i)
      printf("%d\n", *(d + i));
   return 0;
}
```

Output:-

```
enter the value of n:6
Enter elements: 12
34
56
54
32
21
You entered:
12
34
56
54
32
21
```

PRACTICAL-30

Aim-WAP in C to find the factorial of a given number using pointers.

Input:-

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

int main()
{
   Double num,i,f,*p1,*p2;
   p1=&num;
   p2=&f;
   printf("enter any number:");
   scanf("%lf",p1);
   f=1;
   for ( i = 1; i <=*p1; i++)
   {
      *p2=*p2*i;
   }
   printf("\n Factorial of %f is=%f\n",*p1,*p2);
   return 0;
}
```

Output:-

```
enter any number:6
Factorial of 6 is=720
```

PRACTICAL-31

Aim-WAP in C to sort an array using Pointer.

Input:-

```
#include<stdio.h>
#include <stdio.h>
int main()
{
   int *a,i,j,tmp,n;
   printf(" Input the elements to store in the array : ");
   scanf("%d",&n);
   printf(" Input %d  elements in the array : \n",n);
   for(i=0;i<n;i++)
      {
         printf(" element : %d : ",i+1);
         scanf("%d",a+i);
         }
   for(i=0;i<n;i++)
   {
    for(j=i+1;j<n;j++)
    {
       if( *(a+i) > *(a+j))
       {
      tmp = *(a+i);
      *(a+i) = *(a+j);
      *(a+j) = tmp;
       }
    }
   }
   printf("\n The elements in  array after sorting : \n");
   for(i=0;i<n;i++)
      {
         printf(" element : %d : %d \n",i+1,*(a+i));
         }
printf("\n");
return 0;
}
```

Output:-

```
Input the elements to store in the array : 5
 Input 5  elements in the array :
 element : 1 : 21
 element : 2 : 34
 element : 3 : 56
 element : 4 : 78
 element : 5 : 43

 The elements in  array after sorting :
 element : 1 : 21
 element : 2 : 34
 element : 3 : 43
 element : 4 : 56
 element : 5 : 78
```

PRACTICAL-32

Aim-WAP in C to compute the sum of all elements in an array using pointers.

Input:-

```
#include <stdio.h>
int main()
{
  int a[10];
  int i,n, sum = 0;
  int *p;
  printf(" Input the number of elements to store in the array (max 10) : ");
  scanf("%d",&n);
  printf(" Input %d number of elements in the array : \n",n);
  for(i=0;i<n;i++)
    {
        printf(" element - %d : ",i+1);
        scanf("%d",&a[i]);
        }
  p = a; // pt store the base address of array arr1
  for (i = 0; i < n; i++) {
    sum = sum + *p;
    p++;
  }
  printf(" The sum of array is : %d\n\n", sum);
  return 0;
}
```

Output:-

```
 Input the number of elements to store in the array (max 10) : 5
 Input 5 number of elements in the array :
 element - 1 : 1
 element - 2 : 2
 element - 3 : 3
 element - 4 : 4
 element - 5 : 5
 The sum of array is : 15
```

PRACTICAL-33
Aim-WAP in C to print a string in reverse using a pointer.
Input:-

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

int main()
{
        char *s;
        int l,i;
        printf("\nenter a string: ");
        gets(s);
        l=strlen(s);
        printf("\nthe reverse of the string is:");
        for(i=l;i>=0;i--)
                printf("%c",*(s+i));

    return 0;
}
```

Output:-

```
ENTER A STRING: hii

THE REVERSE OF THE STRING IS:iih
```

PRACTICAL-34

Aim-WAP to find length of string entered by user using char pointer.

Input:-

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

int main() {
  char a[100], *p;
  int i = 0;

  printf("Enter Any string  : ");
  gets(a);
  p = a;
  while (*p != '\0') {
    i++;
    p++;
  }
  printf("Length of String : %d", i);

  return 0;
}
```

Output:-

```
Enter Any string  : hello i am aditya
Length of String : 17
```

PRACTICAL-35

Aim-WAP to Find Largest Number Using Dynamic Memory Allocation

Input:-

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

int main() {
    int num;
    float *d;
    printf("Enter the total number of elements: ");
    scanf("%d", &num);
    d = (float *)calloc(num, sizeof(float));
    if (d == NULL) {
        printf("Error!!! memory not allocated.");
        exit(0);
    }
    for (int i = 0; i < num; ++i) {
        printf("Enter Number %d: ", i + 1);
        scanf("%f", d + i);
    }
    for (int i = 1; i < num; ++i) {
        if (*d < *(d + i))
            *d = *(d + i);
    }
    printf("Largest number = %.2f", *d);

    return 0;
}
```

Output:-

```
Enter the total number of elements: 5
Enter Number 1: 21
Enter Number 2: 32
Enter Number 3: 45
Enter Number 4: 34
Enter Number 5: 32
Largest number = 45.00
```

PRACTICAL-36

Aim-WAP to demonstrate wild pointer and constant pointer

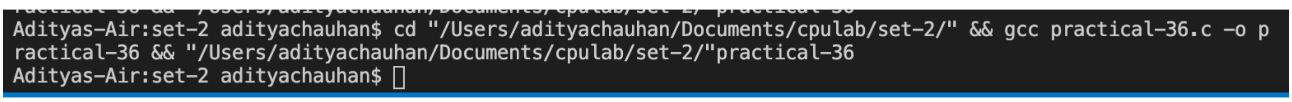
Input:-

```
#include<stdio.h>
int main()
{
  printf("\nWild pointer:\n");
  int *p;
  *p = 12;
/*printf("\nConstant pointer:\n");
int a = 10;
const int* ptr = &a;
*p = 30;*/

return 0;
}
```

Output:-

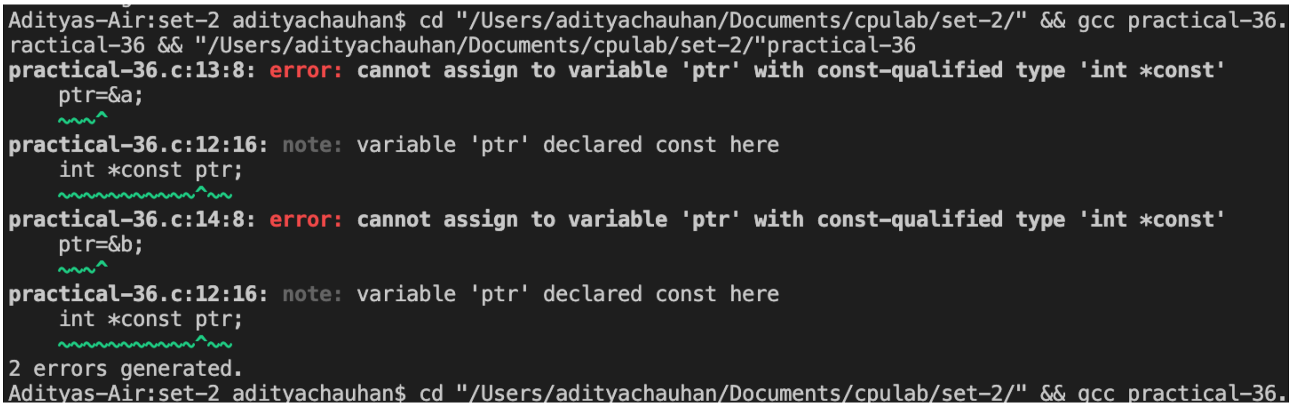
Wild pointer



```
Adityas-Air:set-2 adityachauhan$ cd "/Users/adityachauhan/Documents/cpulab/set-2/" && gcc practical-36.c -o p
ractical-36 && "/Users/adityachauhan/Documents/cpulab/set-2/"practical-36
Adityas-Air:set-2 adityachauhan$ 
```

Constant pointer

```
practical-36.c:13:8: error: cannot assign to variable 'ptr' with const-qualified type 'int
*const'
    ptr=&a;
    ~~~^
practical-36.c:12:16: note: variable 'ptr' declared const here
    int *const ptr;
    ~~~~~~~~~~~^~~
practical-36.c:14:8: error: cannot assign to variable 'ptr' with const-qualified type 'int
*const'
    ptr=&b;
    ~~~^
practical-36.c:12:16: note: variable 'ptr' declared const here
    int *const ptr;
    ~~~~~~~~~~~^~~
2 errors generated.
```



```
Adityas-Air:set-2 adityachauhan$ cd "/Users/adityachauhan/Documents/cpulab/set-2/" && gcc practical-36.
ractical-36 && "/Users/adityachauhan/Documents/cpulab/set-2/"practical-36
practical-36.c:13:8: error: cannot assign to variable 'ptr' with const-qualified type 'int *const'
    ptr=&a;
    ~~~^
practical-36.c:12:16: note: variable 'ptr' declared const here
    int *const ptr;
    ~~~~~~~~~~~^~~
practical-36.c:14:8: error: cannot assign to variable 'ptr' with const-qualified type 'int *const'
    ptr=&b;
    ~~~^
practical-36.c:12:16: note: variable 'ptr' declared const here
    int *const ptr;
    ~~~~~~~~~~~^~~
2 errors generated.
Adityas-Air:set-2 adityachauhan$ cd "/Users/adityachauhan/Documents/cpulab/set-2/" && gcc practical-36.
```

PRACTICAL-37

Aim-WAP to display values and addresses of elements of two dimensional array using pointers.

Input:-

```
#include <stdio.h>
#define ROW     3
#define COL     3
#define TOTAL_CELLS (ROW * COL)
int main(void)
{
   // 2d array
   int i=0,j=0,aiData [ROW][COL] = { { 9, 6, 1 }, { 144, 70, 50 }, {10, 12, 78} };
   printf("\nArray elements: \n");
   printf("\n");
   for(i=0;i<3;i++)
   {
      for(j=0;j<3;j++)
      {
         printf("Array[%d][%d] - %d\n",i+1,j+1,aiData[i][j]);
      }
   }
   int *piData = NULL;
   int arrayIndex = 0;
   piData = &aiData[0][0];
   printf("\nElements printed using pointer: \n");
   for (arrayIndex = 0; arrayIndex < TOTAL_CELLS; ++arrayIndex)
   {
      printf(" array elements = %d\n", *(piData + arrayIndex ));
   }
   return 0;
}
```

Output:-

```
Array elements:

Array[1][1] - 9
Array[1][2] - 6
Array[1][3] - 1
Array[2][1] - 144
Array[2][2] - 70
Array[2][3] - 50
Array[3][1] - 10
Array[3][2] - 12
Array[3][3] - 78

Elements printed using pointer:
 array elements = 9
 array elements = 6
 array elements = 1
 array elements = 144
 array elements = 70
 array elements = 50
 array elements = 10
 array elements = 12
 array elements = 78
```