Experiment 1 :- Array

1. WAP in C to copy the elements of one array into another array.

```
# include <stdio.h>
int main ()
{
    int n;
    printf ("Enter number of elements to input");
    scanf ("%d", &n);
    int a[n] , b[n];
    for (int i=0 ; i<n ; i++)   // to input elements
        scanf ("%d ", a+i);
    for (int i=0; i<n ;i++)   // to copy
        b[i] = a[i];
    for (int i=0; i<n ; i++)       // to print copied array
        printf ("%d ", b[i]);
    return 0 ;
}
```

2. Write a program which takes 10 inputs in an integer array and display their values in the reverse order.

```
# include < stdio.h >
int main ()
{
    int a [10];
    for (int i=0 ; i<10; i++)      // to input
        scanf ("%d ", & a[i]);
    for ( int i= 9 ; i>=0 ; i--)  // to print in reverse order
        printf ("%d ", a[i]);
    return 0;
}
```

3. Write a program to count the number of positive and negative numbers in an array as input by the user.

```
# include <stdio.h>
int main ()
{   int n;
    scanf ("%d", &n);
    int a[n];
    int p=0 , ne=0;
    for (int i=0 ; i<n ; i++)
    {
        scanf ("%d", &a[i]);
        if (a[i] >=0)  p++;
        else  ne++;
    }
    printf ("positive = %d \n negative = %d", p, ne);
    return 0;
 }
```

4. Write a program in C to separate odd and even integers in separate arrays.

```
# include <stdio.h>
int  main ()
{  int n;
   scanf ("%d", &n);
   int a[n], odd[n], even[n], e=0, d=0;
   printf (" Enter elements \n");
   for (int i=0; i<n ; i++)
   {
       scanf ("%d", &a[i]);
```

```
      if (a[i] % 2 ==0)
               even[e++]=a[i];
      else     odd[d++] = a[i];
    }
    printf (" even => ");
    for (int i=0; i<e; i++)
         printf ("%d ", even[i]);
    printf ("\n odd => ");
    for (int i=0; i<d; i++)
         printf ("%d ", odd[i]);
    return 0;
}
```

Experiment 2 :- 2 D Array

1. Take input from the user in a 2D array and print the row-wise and column-wise sum of numbers stored in this 2 D array.

```
# include <stdio.h>
int main ()
{
    int m,n,i,j;
    scanf ("%d %d", &m, &n);
    int a[m][n]; int s=0;
    for (i=0; i<m; i++)
        for (j=0; j<n; j++)
            scanf ("%d", &a[i][j]);
  for (i=0; i<m; i++)
  {
      for (j=0; j<n; j++)
      s = s + a[i][j];
      printf ("sum of row %d=%d\n", i+1, s);
      s=0;
  }
  for (i=0; i<n; i++)
  {   for (j=0; j<m; j++)  s=s + a[j][i];
      printf ("sum of column %d =%d \n", i+1, s);
      s=0;
  }
  return 0;
}
```

2. Write a program to multiply two Matrices.

```
# include <stdio.h>
int main ()
{  int m,n,p,q; int i,j;
   scanf ("%d %d %d %d", &m, &n, &p, &q);
   int a[m][n], b[p][q], c[m][q];
   printf ("Enter 1st matrix \n");
   for (i=0; i<m; i++)
        for (j=0; j<m; j++)
              scanf ("%d", &a[i][j]);
   printf ("Enter 2nd matrix \n");
   for (i=0; i<p; i++)
        for (j=0; j<q; j++)
              scanf ("%d", & b[i][j]);
   if (n==p)
   {
     for (i=0; i<m; i++)
     {
        for (j=0; j<q; j++)
        {
           int s=0;
           for (int r=0; r<n; r++)
                s= s + a[i][r] * b[r][j];
           c[i][j]= s;
        }
     }  printf ("A x B=\n");
     for (i=0; i<m; i++)
     {
         for (j=0; j<q; j++)
              printf ("%d \t", c[i][j]);
         printf ("\n");
     }
```

```
    }
    else
    printf("cant Multiply");
    return 0;
}
```

3. Write a program in C to calculate determinant of a 3×3 matrix.

```
# include <stdio.h>
int  main ()
{
     int i, j .s=0;  int a[3][3];
     for (i=0 ; i<3 ; i++)
           for (j=0 ; j<3 ; j++)
                  scanf("%d", & a[i][j]);
     for (i=0 ; i<3 ; i++)
              s= s+ a[0][i] * (a[1][(i+1)%3]*a[2][(i+2)%3] -
                   a[1][(i+2)%3] * a[2][(i+1)%3]);
     printf("Determinant = %d", s);
     return 0;
 }
```

4. Write a program in C to check whether a given matrix is an identity matrix.

```
# include <stdio.h>
int main ()
{
     int m, n, i, j;
     scanf("%d %d", &m. &n);
     int a[m][n]; int f=0;
if(m==n)
{
        for (i=0;i<m;i++)
        {
                for (j=0; j<n; j++)
                {
                        scanf("%d", &a[i][j]);
                        if(i==j && a[i][j]!=1) f=1;
                        else if(i!=j && a[i][j]!=0) f=1;
                }
        }
        if(f==1) printf("Not identity Matrix");
        else printf("Identity Matrix");
}
else
printf("Matrix must be square matrix");
return 0;
}
```

Experiment 3 :- String

1. Write a program to check whether given string is palindrome or not.

```
# include <stdio.h>
# include <string.h>
int main ()
{
    char s [10000];
    gets (s); char s2[10000]; strcpy (s2, s);
    strrev (s2);
    if (strcmp (s2, s)==0) printf ("palindrome");
    else printf ("not palindrome");

    return 0;
}
```

2. Write a program to calculate total number of consonants, vowels and other characters in a given string.

```
# include <stdio.h>
# include <string.h>
int main ()
{
     char s [10000]; gets (s); int i, v=0, c=0, d=0;
     for (i=0; s[i]!=0 ;i++)
     {
        if(s[i]>=65 && s[i] <=90 || s[i] >=97 && s[i] <= 'z')
        {
           if ( s[i]=='A' || s[i]=='E'|| s[i]=='O' || s[i]=='I' ||
                s[i]=='U' || s[i]=='a' || s[i]=='e' || s[i]=='i' ||
                s[i]=='o' || s[i] == 'u') v++;
           else c++;
        }
        else  d++;
    }
    printf("vowels = %d \n consonants = %d \n others = %d", v,c,d);
    return 0;
}
```

3. Write a program to input a word from the user and print it in the following way. For example, if the word is PROGRAM, the program will print it as –

```
P
P R
P R O
P R O G
P R O G R
P R O G R A
P R O G R A M
```

```
# include <stdio.h>
# include <string.h>
int main ()
{
    char s[10000]; gets (s);
    int l = strlen (s);
    for ( int i=0; i<l; i++)
    {
        for ( int j=0; j<=i; j++)
            printf("%c". s[j]);
        printf("\n");
    }
    return 0;
}
```

4. Write a program to search a middle name in the name consisting of first name, middle name and last name.

```
# include <stdio.h>
# include <string.h>
int main ()
{
      char s[10000]; gets (s);
      char *p=strchr (s,' ');
      while (* ++p != ' ');
               printf ("%c", *p);
      return 0;
}
```

Experiment 4 :- Function

1. Write a program to calculate x^n without using library function pow ( ).

```
# include <stdio.h>   int power (int , int);
int main ( )
{
      int  x , n ; int p ;
      scanf (" %d %d ", &x , &n);
    p=power (x , n) ;
    printf (" %d ", p) ;
    return  0 ;
}
int  power( int x,  int n )
 {
        int  p =1 ;
        while (n--)
                P =P*X ;
        return  p ;
 }
```

2. Write a program that input the meal charge of a customer. The tax should be 20% of the meal cost. The tip should be 15% of the total after adding the tax. Display the total bill on the screen.

```
# include <stdio.h>
void  tb (float  m)
{
      m =  m + 20*m/100 ;    //adding  tax
      m =  m  + 15* m/100 ;  // adding  tip
    printf (" Total bill = Rs. %.2f ", m) ;
}
int  main ( )
{   float m ; scanf(" %f ", &m); tb(m) ;
    return 0 ;
}
```

3. Write a program to input coefficients of quadratic equation and pass them to function () QUAD. This returnable function computes whether roots of a quadratic equation are real or imaginary.

```
# include <stdio.h>
float QUAD (float a, float b, float c)
 {
      float d= b*b -4*a*c;
      if (d>=0)
              return 1;
      else    return -1;
 }
int main ()
 {
      float a, b, c, d;
      scanf ("%f %f %f", &a, &b, &c);
      d= QUAD (a, b, c);
      if (d==1) printf ("Roots are real");
      else  printf ("Roots are imaginary");
      return 0;
 }
```

4. Write a program to calculate binomial coefficient.

```
  # include <stdio.h>
  int fact (int n)
  {
       int f=1;
       while (n--)
            f=f*n;
       return f;
  }
```

```
int main ()
{
    int n, r;
    scanf ("%d %d", &n, &r);
    int c = fact(n) / (fact(r) * fact(n-r));
    printf ("Binomial coefficient = %d", c);
    return 0;
}
```

Experiment 5:- Pointer

1. WAP to search an element in an array using pointers.

```
# include <stdio.h>
int main ()
{
      int n,i,j; scanf("%d", &n);
      int a[n], *p=a;
      for(i=0; i<n; i++)
              scanf("%d", p+i);
      int e, *q=&e;
      scanf("%d", q);
      int f=0;
      for(i=0; i<n; i++)
      {
            if (*(p+i)==*q)
            { f=1;
                 printf("present at position %d", i+1);
            }
      }
      if (f==0)
            printf("Not found");
      return 0;
}
```

2. WAP that print the position of largest element in an array using pointer.

```
# include <stdio.h>      # include <math.h>
int  main ( )
{
      int  n, i;
      scanf ( "%d", &n );
      int  a[n], *p = a ;
      int  max = -pow (2,31) , t ;
      for (i=0 ; i<n ; i++)
      {
            scanf ("%d", p+i);
            if (*(p+i) > max)
            {
                  max = *(p+i) ;
                  t = i+1 ;
            }
      }
   printf ("largest element in array is at position %d", t);
   return 0;
}
```

3. WAP to categorize each element of an array as prime or not using pointer.

```
# include <stdio.h>
int  main ( )
{
     int  n;
     scanf( "%d", & n);
int a[n], *p=a;
int i, j;
for(i=0; i<n; i++)
        scanf("%d", p+i);
for(i=0; i<n; i++)
{
        int c=0;
        for (j=1; j<=*(p+i); j++)
                if (*(p+i) %j==0)
                        c++;
        if (c==2)
                printf("%d is prime \n", p[i]);
        else printf("%d is not prime \n", *(p+i));
}
return 0;
}
```

4. There are two students Rita and Sita, store their 5 subject marks in two different array. Write a C program to find who scored more in individual subject as well as in averege using pointer.

```
#include <stdio.h>
int main ()
{
    int R[5], S[5], *p=R, *q=S;
    int i, sumr=0 , sums=0;
    float avgr, avgs;
    for (i=0; i<5; i++)
    {
        printf("Enter marks of Rita and Sita in subject %d", i+1);
        scanf("%d %d", p+i, q+i);
        sumr = sumr + *(p+i);
        sums = sums + *(q+i);
    }
    avgr = sumr/5.0;   avgs = sums/5.0;
    for (i=0; i<5; i++)
    {
        if (*(p+i) > *(q+i))
            printf("Ria scored more in subject %d \n", i+1);
        else if (*(q+i) > *(p+i))
            printf("Sia scored more in subject %d \n", i+1);
        else printf("Both scored same in subject %d \n", i+1);
    }
    if (avgr > avgs)
        printf("Ria scored more in average \n");
    else if (avgs > avgr)
        printf("Sia scored more in average \n");
    else printf(" Both have same average\n");
    return 0;
}
```