1. Write a C program to take N numbers as an input in an array and print them.

```
#include<stdio.h>
int main()
{
    int a[1000],i,n;

     printf("Enter size of array: ");
    scanf("%d",&n);

     printf("Enter %d elements in the array : ", n);
    for(i=0;i<n;i++)
    {
        scanf("%d", &a[i]);
    }

    printf("\nElements in array are: ");
    for(i=0;i<n;i++)

    {
        printf("%d  ", a[i]);
    }

    return 0;
}
```

```
Output:
Enter size of array: 3
Enter 3 elements in the array : 1
2
3

Elements in array are: 1  2  3
```

2. Write a C program to insert a number in a given position in an array.

```
#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
3
Enter 3 elements
1
2
3
Enter the location where you wish to insert an element
2
Enter the value to insert
4
Resultant array is
1
4
2
3
```

3. Write a C program to delete a number in a given position in an array.

```
#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
3
Enter 3 elements
1
2
4
Enter the location where you wish to delete element
2
Resultant array:
1
4
```

4. Write a C program to search a number in an array and also print the position of the input number.

```
#include <stdio.h>

int main()

{
   int arr[250], search, n, i;

   printf("Please enter how many elements should be available in an
array\n");

   scanf("%d",&n);

  printf("\nPlease enter %d numbers or integers one by one", n);

   for (i = 0; i < n; i++)

      scanf("%d", &arr[i]);

   printf("\nPlease enter the number you want to search");

   scanf("%d", &search);

 for (i = 0; i < n; i++)
   {
       if (arr[i] == search)
      {
         printf("\n%d is present at location %d\n", search, i+1);
         break;
      }
   }
```

```
   if (i == n)

      printf("%d is not available in the array.\n", search);

   return 0;

}
```

```
Output:
Please enter how many elements should be available in an array
3

Please enter 3 numbers or integers one by one1
2
3

Please enter the number you want to search2

2 is present at location 2
```

5. Write a C program to sort an array element.

```
#include <stdio.h>
int main()
{
    int arr[] = {5, 2, 8, 7, 1};
    int temp = 0;
  int length = sizeof(arr)/sizeof(arr[0]);
    printf("Elements of original array: \n");
    for (int i = 0; i < length; i++) {
        printf("%d ", arr[i]);
    }
    for (int i = 0; i < length; i++) {
        for (int j = i+1; j < length; j++) {
           if(arr[i] > arr[j]) {
               temp = arr[i];
               arr[i] = arr[j];
               arr[j] = temp;
           }
        }
    }

    printf("\n");
    printf("Elements of array sorted in ascending order: \n");
    for (int i = 0; i < length; i++) {
        printf("%d ", arr[i]);
    }
    return 0;
}
```

```
Output:

Elements of original array:
5 2 8 7 1
```

```
Elements of array sorted in ascending order:
1 2 5 7 8
```

6. Write a C program to print the address of a given input.

```
#include<stdio.h>
int main( )
{
        int arr[5];
        int i;
        printf("Enter the array 5 elements : ");
        for(i=0; i<5; i++)
        {
                scanf("%d", &arr[i]);
        }
        printf("\nArray elements with their addresses using pointers :
\n");

        for(i=0; i<5; i++)
        {
                printf("Value of arr[%d] = %d\t", i,*(arr+i));
                printf("Address of arr[%d] = %p\n",i,arr+i);
        }

        return 0;
}
```

```
Output:
Enter the array 5 elements :
6
2
9
1
7

Array elements with their addresses using pointers :

Value of arr[0] = 6     Address of arr[0] = 000000000062FE30
Value of arr[1] = 2     Address of arr[1] = 000000000062FE34
Value of arr[2] = 9     Address of arr[2] = 000000000062FE38
Value of arr[3] = 1     Address of arr[3] = 000000000062FE3C
Value of arr[4] = 7     Address of arr[4] = 000000000062FE40
```

7. Write a C program to count the number of vowel and consonant in a character array.

```
#include <string.h>

int main()
{
    char s[1000];
    int i,vowels=0,consonants=0;

    printf("Enter  the string : ");
    gets(s);
```

```
    for(i=0;s[i];i++)
    {
      if((s[i]>=65 && s[i]<=90)|| (s[i]>=97 && s[i]<=122))
      {

            if(s[i]=='a'||
s[i]=='e'||s[i]=='i'||s[i]=='o'||s[i]=='u'||s[i]=='A'||s[i]=='E'||s[i]=='
I'||s[i]=='O' ||s[i]=='U')
                  vowels++;
            else
             consonants++;
        }

      }


    printf("vowels = %d\n",vowels);
    printf("consonants = %d\n",consonants);

    return 0;
}

Output:
Enter  the string : sayan
vowels = 2
consonants = 3


8. Write a C program to take M X N matrix as an input and print the
matrix properly.

#include<stdio.h>

int main()
{
    int array1[10][10],i,j,m,n,sum = 0;

    printf("Enter no. of rows :: ");
        scanf("%d", &m);
        printf("\nEnter no. of cols :: ");
        scanf("%d",&n);
        printf("\nEnter values to the matrix :: \n");
        for (i = 0; i < m; i++)
        {
            for (j = 0; j < n; j++)
            {
                 printf("\nEnter a[%d][%d] value :: ",i,j);
                 scanf("%d", &array1[i][j]);
        }
    }

    printf("\nThe given matrix is \n\n");
        for (i = 0; i < m; ++i)
        {
        for (j = 0; j < n; ++j)
        {
            printf("\t%d", array1[i][j]);
```

```
        }
        printf("\n\n");
    }

    return 0;
}
```

Output:

```
Enter no. of rows :: 3

Enter no. of cols :: 3

Enter values to the matrix ::

Enter a[0][0] value :: 9

Enter a[0][1] value :: 8

Enter a[0][2] value :: 7

Enter a[1][0] value :: 6

Enter a[1][1] value :: 5

Enter a[1][2] value :: 4

Enter a[2][0] value :: 3

Enter a[2][1] value :: 2

Enter a[2][2] value :: 1

The given matrix is

        9       8       7

        6       5       4

        3       2       1
```

9. Write a C program to addition and subtraction of two matrices.

```
#include<stdio.h>
int main()
{
  int r, c, a[100][100], b[100][100], sum[100][100],diff[100][100],i, j;
  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);

  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]);
    }

  printf("Enter elements of 2nd matrix:\n");
  for (i = 0; i < r; ++i)
    for (j = 0; j < c; ++j) {
      printf("Enter element b%d%d: ", i + 1, j + 1);
      scanf("%d", &b[i][j]);
    }


  for (i = 0; i < r; ++i)
    for (j = 0; j < c; ++j) {
      sum[i][j] = a[i][j] + b[i][j];
    }

  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");
      }
    }

       for (i = 0; i < r; ++i)
    for (j = 0; j < c; ++j) {
      diff[i][j] = a[i][j] - b[i][j];
    }

  printf("\nDifferencs of two matrices: \n");
  for (i = 0; i < r; ++i)
    for (j = 0; j < c; ++j) {
      printf("%d   ", diff[i][j]);
      if (j == c - 1) {
        printf("\n\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: 1
Enter element a12: 1
Enter element a21: 2
Enter element a22: 3
Enter elements of 2nd matrix:
Enter element b11: 4
Enter element b12: 5
Enter element b21: 3
Enter element b22: 2
```

```
Sum of two matrices:
5   6

5   5


Differencs of two matrices:
-3   -4

-1   1
```

10. Write a C program to transpose of a matrix.

```
#include <stdio.h>
int main()
 {
  int a[10][10], transpose[10][10], r, c;
  printf("Enter rows and columns: ");
  scanf("%d %d", &r, &c);


  printf("\nEnter matrix elements:\n");
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    printf("Enter element a%d%d: ", i + 1, j + 1);
    scanf("%d", &a[i][j]);
  }


  printf("\nEntered matrix: \n");
  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    printf("%d  ", a[i][j]);
    if (j == c - 1)
    printf("\n");
  }


  for (int i = 0; i < r; ++i)
  for (int j = 0; j < c; ++j) {
    transpose[j][i] = a[i][j];
  }


  printf("\nTranspose of the matrix:\n");
  for (int i = 0; i < c; ++i)
  for (int j = 0; j < r; ++j) {
    printf("%d  ", transpose[i][j]);
    if (j == r - 1)
    printf("\n");
  }
  return 0;
}
```

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

```
3

Enter matrix elements:
Enter element a11: 1
Enter element a12: 4
Enter element a13: 0
Enter element a21: -5
Enter element a22: 2
Enter element a23: 7

Entered matrix:
1  4  0
-5  2  7

Transpose of the matrix:
1  -5
4  2
0  7
```

11. Write a C program to multiplication of two matrices.

```
#include <stdio.h>

int main()
{
  int m, n, p, q, c, d, k, sum = 0;
  int first[10][10], second[10][10], multiply[10][10];

  printf("Enter the number of rows and columns of first matrix\n");
  scanf("%d%d", &m, &n);
  printf("Enter the elements of first matrix\n");

  for (  c = 0 ; c < m ; c++ )
    for ( d = 0 ; d < n ; d++ )
      scanf("%d", &first[c][d]);

  printf("Enter the number of rows and columns of second matrix\n");
  scanf("%d%d", &p, &q);

  if ( n != p )
    printf("Matrices with entered orders can't be multiplied with each
other.\n");
  else
  {
    printf("Enter the elements of second matrix\n");

    for ( c = 0 ; c < p ; c++ )
      for ( d = 0 ; d < q ; d++ )
        scanf("%d", &second[c][d]);

    for ( c = 0 ; c < m ; c++ )
    {
      for ( d = 0 ; d < q ; d++ )
      {
        for ( k = 0 ; k < p ; k++ )
        {
          sum = sum + first[c][k]*second[k][d];
        }
```

```
        multiply[c][d] = sum;
        sum = 0;
      }
    }

    printf("Product of entered matrices:-\n");

    for ( c = 0 ; c < m ; c++ )
    {
      for ( d = 0 ; d < q ; d++ )
        printf("%d\t", multiply[c][d]);

      printf("\n");
    }
  }

  return 0;
}
```

```
Output:
Enter the number of rows and columns of first matrix 3 3
Enter the elements of first matrix
1 2 0
0 1 1
2 0 1
Enter the number of rows and columns of second matrix 3 3
Enter the elements of second matrix
1 1 2
2 1 1
1 2 1
Product of entered matrices:-
5     3     4
3     3     2
3     4     5
```

12. Write a C program to addition and subtraction of two matrices (using pointer and malloc() function).

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

void main()
{
    int **X, **Y, **sum, **diff, m, n, i, j;
    printf("Enter number of rows in the matrices: ");
    scanf("%d",&m);
    X=(int **)malloc(m*sizeof(int));
    Y=(int **)malloc(m*sizeof(int));
    sum=(int **)malloc(m*sizeof(int));
    diff=(int **)malloc(m*sizeof(int));
```

```
    printf("Enter number of columns in the matrices: ");
    scanf("%d",&n);

    for (i = 0; i < m; i++)
    {
        X[i] = (int *) malloc(n * sizeof(int));
        Y[i] = (int *) malloc(n * sizeof(int));
        sum[i] = (int *) malloc(n * sizeof(int));
        diff[i] = (int *) malloc(n * sizeof(int));
    }

    printf("\nEnter first matrix: \n");
    for (i=0;i<m;i++)
        for (j=0;j<n;j++)
            scanf("%d", &X[i][j]);

    printf("\nEnter second matrix: \n");
    for (i=0;i<m;i++)
        for (j=0;j<n;j++)
            scanf("%d", &Y[i][j]);

    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            (*(*(sum+i)+j))= (*(*(X+i)+j)) + (*(*(Y+i)+j));

    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            (*(*(diff+i)+j))= (*(*(X+i)+j)) - (*(*(Y+i)+j));

    printf("\nSum of matrices: \n");
    for (i=0;i<m;i++)
    {
        for (j=0;j<n;j++)
            printf("%d\t", sum[i][j]);
        printf("\n");
    }

    printf("\nDifference of matrices:\n");
    for (i=0;i<m;i++)
    {
        for (j=0;j<n;j++)
            printf("%d\t", diff[i][j]);
        printf("\n");
    }
}
```

Output:

```
Enter number of rows in the matrices: 2
Enter number of columns in the matrices: 2

Enter first matrix:
1
2
3
4
```

```
Enter second matrix:
1
2
3
4

Sum of matrices:
2       4
6       8

Difference of matrices:
0       0
0       0
```