Unit 4

Arrays and Strings

Array

What is array?

Array is a set of similar data elements which are stored in consecutive memory location under a common variable name.

So to be an array,

- All elements must be of same data type
- All elements are stored in consecutive memory location.

Eg. If we want to store 100 integers in a sequence, we can create array for it.

int num[100];

Note: *The size and type of array cannot be changed after its declaration.*

Why do we use array in programming language?

If we want many elements of similar type than it is not feasible to declare all variables and also manipulate these elements. So in this case we use array.

For example, if we want 100 integer variables, then instead of writing all 100 variables, we use array like int num[100]. Here 100 integer elements num[0], num[1], num[2]. . . . . .num[99] are declared.

Limitation of Array

Memory allocation in array are static in nature. Which means memory is allocated before the execution of program begins (During compilation).In this type of allocation the memory cannot be resized after initial allocation. So it has some limitations.

- Wastage of memory
- Overflow of memory

eg. int num[100];
Here, the size of an array has been fixed to 100.If we just enter to 10 elements only, then their will be wastage of 90 memory location and if we need to store more than 100 elements there will be memory overflow.

Why array is called static data type?

The array is called static data type because when the size of array is once allocated,it cannot be modified.So that maximum size needed must known in advance.

eg. int num[100];

Here, the size of an array has been fixed to 100 and cannot be chage during program execution.

Types of Array

1. One dimensional Array
2. Multidimensional Array

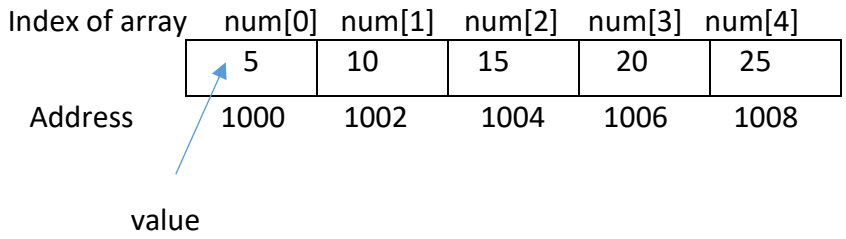
One dimensional array

Elements of an array can be represented either as a single row or single column.

There is a single subscript or index whose value refers to the individual array element, which ranges from 0 to n-1, where n is the size of array.

Eg. int num[5]={5,10,15,20,25};

Assuming base address 1000, now array elements can be illustrated as:



Declaration:

data_type array_name[size];

eg. int num[5];

Here, int is a data type and num is the name of the array and 5 is the size of the array. It means num can only contain 5 elements of int type.

Initialization of 1-D array

How can you initialize one dimensional array at compile time and run time? Explain with suitable example.

After the array is declared it must be initialized, otherwise it will contain garbage value (any random value). An array can be initialized at either compile time or runtime.

1) Compile time initialization

General form:

data_type array_name[size]={list of values};

The values in the list are separated by commas.

Example:

```
int  num[5]={5,10,15,20,25};          //integer array initialization
float area[5]={33.4,55.6,88.9,77.8,5.5};      //float array initialization
```

Note: It is also possible to initialize array without defining its size.

```
int num[]={67,87,56,24,77};
```

In this case compiler determines the size of array by calculating the number of elements in an array.

Accessing Array Elements

We can access the elements of an array by giving the name and proper subscript inside the bracket.

Eg. int num[5]={5,10,15,20,25};

Assuming base address 1000, now array elements can be illustrated as:

Index of array	num[0]	num[1]	num[2]	num[3]	num[4]
	5	10	15	20	25
Address	1000	1002	1004	1006	1008

Here, num[0] is 5 which is stored at 1000.

Similarly,
num[1]=10,
num[2]=15,
num[3]=20,
num[4]=25

Example

```
#include<stdio.h>
int main()
{
int i;
int  num[5]={5,10,15,20,25};
for(i=0;i<5;i++)
{
printf("%d\t",num[i]);
}
return 0;
}
```

Output

5 10 15 20 25

2) Runtime initialization

An array can also be initialized at runtime using scanf() function. This Approach is used for initializing large arrays with user specified values.

```
#include<stdio.h>
int main()
{
int num[5];
int i;
printf("Enter the array elements\n");
for(i=0;i<5;i++)
{
 scanf("%d",&num[i]);
}
printf("Array elements are\n");
for(i=0;i<5;i++)
{
 printf("%d\n",num[i]);
}
return 0;
}
```

Multidimensional Array

Multidimensional array are those who have two or more than two dimensions.

Multidimensional array are defined in the same manner as one dimensional arrays, except that separate pair of square brackets is required for each subscript.

Thus, two dimensional array will require two pair of square brackets, a three dimensional array will require three pairs of square brackets and so on.

General form:

datatype array_name[s1][s2][s3]….[sn]

where, si is the size of ith dimension.

Example:

int survey[3][5][12];

Here, survey is a three dimensional array declared to contain 180 integer type elements.

Two-dimensional array

- A two-dimensional array is a collection of similar data items, structured in two dimensions (referred to as rows and columns).
- It is also called array of arrays.
- It is required to manipulate the data in table format or in matrix format which contains rows and columns.

Declaration:

data_type array_name[row_size][column_size];

eg. int arr[3][3];

It creates two dimensional array to store 9 elements of integer type. There are 3 rows and 3 columns in array matrix.

	Column 1	Column 2	Column 3
Row1	arr[0][0]	arr[0][1]	arr[0][2]
Row2	arr[1][0]	arr[1][1]	arr[1][2]
Row 3	arr[2][0]	arr[2][1]	arr[2][2]

Initialization of 2-D array

1) Compile time initialization

Two dimensional array may be initialized by following their declaration with a list of values enclosed in braces.

Example: int arr[3][3]={ {2,4,6},{8,9,12},{15,16,18}};

These are equivalent to following assignments

arr[0][0]= 2 arr[0][1]= 4 arr[0][2]= 6

arr[1][0]= 8 arr[1][1]= 9 arr[1][2]= 12

arr[2][0]=15 arr[2][1]= 16 arr[2][2]=18

Similarly, for,

int arr[2][3]={{1,5,10},{1,10,15}};

int arr[3][4]={{65,85,75,50},{67,65,45,75},{35,5,60,50}};

***Note**: When array is completely initialized with all values, explicitly, we need not specify the size of the first dimension.*

That is the statement,

int arr[ ][3]={{2,4,6},{8,9,12},{15,16,18}}; is permitted.

Example:

```
#include<stdio.h>
int main()
{
  int i,j;
  int arr[3][3]={{12,14,16},{5,7,15},{15,25,45}};
  printf("The  Elements of the matrix are\n");
  for(i=0;i<3;i++)
  {
          for(j=0;j<3;j++)
          {
                  printf("%d\t",arr[i][j]);
          }
          printf("\n");
  }
  return 0;
}
```

2) Runtime initialization

Program to input 3*3 matrix and display it.

```
#include<stdio.h>
int main()
{
  int i,j;
  int arr[3][3];
  printf("Enter the elements of matrix\n");
  for(i=0;i<3;i++)
  {
          for(j=0;j<3;j++)
          {
                  scanf("%d",&arr[i][j]);
          }
  }
  printf("Elements of matrix are\n");
  for(i=0;i<3;i++)
  {
          for(j=0;j<3;j++)
          {
                  printf("%d\t",arr[i][j]);
          }
          printf("\n");
  }
  return 0;       }
```

Why array is important in programming? How can you initialize different types of arrays? Explain 2-dimensional array in C.

If we want many elements of similar type than it is not feasible to declare all variables and also manipulate these elements. So in this case we use array.

For example, if we want 100 integer variables, then instead of writing all 100 variables, we use array like int num[100]. Here 100 integer elements num[0], num[1], num[2]. . . . . .num[99] are declared. Due to this reason array is important in programming.

Initialization of 1-D array

1) **Compile time Initialization**

General form: data_type array_name[size]={list of values};

The value in the list are separated by commas.

Example:

```
int  num[5]={10,20,30,40,50};          //integer array initialization
```

2) **Runtime initialization**

An array can also be initialized at runtime using scanf() function. This Approach is used for initializing large arrays with user specified values.

Example:

```
for(i=0;i<5;i++)
{
 scanf("%d",&num[i]);
}
```

This program statement initializes 5 arrays elements num[0],num[1],num[2],num[3],num[4] with user specified values.

Initialization of 2-D array

1) **Compile time initialization**

   Two dimensional array may be initialized by following their declaration with a list of values enclosed in braces.

   Example: int arr[3][3]={ {2,4,6},{8,9,12},{15,16,18}};

   These are equivalent to following assignments

arr[0][0]= 2	arr[0][1]=	4	arr[0][2]= 6
arr[1][0]= 8	arr[1][1]=	9	arr[1][2]= 12
arr[2][0]=15	arr[2][1]=	16	arr[2][2]=18

2) Runtime initialization

Runtime initialization of 2-D array is done by using scanf() function with the concept of rows and columns.

Example

```
for(i=0;i<3;i++)
{
        for(j=0;j<3;j++)
        {
                scanf("%d",&arr[i][j]);
        }
}
```

This statement initializes 9 elements of matrix of size 3*3.

Two dimensional array

- A two dimensional array is a collection of similar data items, structured in two dimensions (referred to as rows and columns).
- It is also called array of arrays.
- It is required to manipulate the data in table format or in matrix format which contains rows and columns.

Declaration:

data_type array_name[row_size][column_size];

eg. int arr[3][3];

It creates two dimensional array to store 9 elements of integer type. There are 3 rows and 3 columns in array matrix.

	Column 1	Column 2	Column 3
Row1	arr[0][0]	arr0][1]	arr[0][2]
Row2	arr[1][0]	arr[1][1]	arr[1][2]
Row 3	arr[2][0]	arr[2][1]	arr[2][2]

Programs related to one dimensional Array

1) Write a program to read 10 elements in an array and display it.

```
#include<stdio.h>
int main()
{
    int i,num[10];
    printf("Enter 10 array elements\n");
    for(i=0;i<10;i++)
     {
       scanf("%d",&num[i]);
     }
     printf("The array elements are\n");
     for(i=0;i<10;i++)
     {
           printf("%d\n",num[i]);
      }
            return 0;
 }
```

2) Write a program to read n elements in array and display it.

```
 #include<stdio.h>
int main()
{
   int i,num[100],n;
   printf("Enter number of array elements \n");
   scanf("%d",&n);
   printf("Enter %d array elements\n",n);
     for(i=0;i<n;i++)
     {
      scanf("%d",&num[i]);
     }
    printf("The array elements are\n");
    for(i=0;i<n;i++)
     {
          printf("%d\n",num[i]);
     }
    return 0;
}
```

3) Write a program to read n elements in an array and display them in reverse order.

```
#include<stdio.h>
 int main()
 {
  int i,num[100],n;
 printf("Enter the  number of  array elements\n");
  scanf("%d",&n);
  printf("Enter %d array elements\n",n);
  for(i=0;i<n;i++)
  {
   scanf("%d",&num[i]);
 }
printf("The array elements in reverse order are\n");
  for(i=n-1;i>=0;i--)
  {
   printf("%d\n",num[i]);
  }
  return 0;
 }
```

4) Write a program to read n elements in an array and find sum of all elements.

```
#include<stdio.h>
int main()
{
   int num[100],n,i,sum=0;
   printf("Enter number of array elements \n");
   scanf("%d",&n);
   printf("Enter %d array elements\n",n);
    for(i=0;i<n;i++)
    {
    scanf("%d",&num[i]);
    }
    for(i=0;i<n;i++)
    {
          sum=sum+num[i];
     }
    printf("sum of array elements=%d\n",sum);
    return 0;
}
```

5) WAP to input n elements in an array and find the sum of all even elements

```
#include<stdio.h>
 int main()
  {
int  num[100],i,n,esum=0;
 printf("Enter number of array elements\n");
scanf("%d",&n);
printf("Enter %d elements\n",n);
 for(i=0;i<n;i++)
  {
   scanf("%d",&num[i]);
  }
  for(i=0;i<n;i++)
   {
    if(num[i]%2==0)
    {
      esum=esum+num[i];
   }
   }
printf("Sum of even elements=%d\n",esum);
return 0;
 }
```

6) WAP to input n elements in an array and find the sum of both odd and even elements separately as well count them.

```
#include<stdio.h>
int main()
{
int num[100],i,n,esum=0,ecount=0,osum=0,ocount=0;
printf("Enter the number of array elements\n");
scanf("%d",&n);
printf("Enter %d elements\n",n);
for(i=0;i<n;i++)
{
 scanf("%d",&num[i]);
}
for(i=0;i<n;i++)
{
    if(num[i]%2==0)
  {
          esum=esum+num[i];
          ecount++;
 }
  else
 {
 osum=osum+num[i];
  ocount++;
 }
}
printf("Sum of Even elements =%d\n",esum);
printf("Number of Even elements=%d\n",ecount);
printf("Sum of Odd elements =%d\n",osum);
printf("Number of Odd elements=%d\n",ocount);
return 0;
}
```

Write a program to input n elements in an array and find their average.

Hint:

```
avg=sum/n;
```

Write a program to input n elements in an array and find the sum of all odd elements.

Hint :

```
for(i=0;i<n;i++)
 {
  if(num[i]%2!=0)
  {
    osum=osum+num[i];
 }
 }
```

7) **WAP to read the marks of 10 students calculate and display the average marks and find deviation of marks of each student from average marks.**

```
#include<stdio.h>
int main()
{
 float marks[10],avg,deviation,sum=0;
 int i;
 printf("Enter the marks of 10 students\n");
 for(i=0;i<10;i++)
 {
  scanf("%f",&marks[i]);
 }
 for(i=0;i<10;i++)
 {
        sum=sum+marks[i];
 }
  avg=sum/10;
  printf("Average marks =%f\n",avg);
  printf("\nDeviation marks of each student from  average marks\n");
  for(i=0;i<10;i++)
  {   deviation=marks[i]-avg;
       printf("marks[%d]=%f\tdeviation=%f\n",i,marks[i],deviation);
   }
  return 0;
}
```

8) WAP to read n numbers in an array find the largest number among them.

```
#include<stdio.h>
int main()
{
int num[100],i,n,large;
printf("Enter the number of array elements\n");
scanf("%d",&n);
printf("Enter %d elements\n",n);
for(i=0;i<n;i++)
{
  scanf("%d",&num[i]);
}
large=num[0];
for(i=1;i<n;i++)
 {
          if(num[i]>large)
          {
            large=num[i];
          }
 }
printf("Largest number=%d",large);
return 0;
 }
```

Assignment:

Write a Program to read n numbers in an array find the smallest number among them.

Hint:

```
small=num[0];
for(i=1;i<n;i++)
 {
          if(num[i]<small)
          {
            small=num[i];
          }
 }
```

9) Write a program to read n number from keyboard and find the smallest and largest number using array.

```
#include<stdio.h>
int main()
{
int num[100],n,i,large,small;
printf("Enter the number of array elements\n");
 scanf("%d",&n);
 printf("Enter %d elements\n",n);
 for(i=0;i<n;i++)
  {
        scanf("%d",&num[i]);
  }
 large=num[0];
 small=num[0];
  for(i=1;i<n;i++)
   {
        if(num[i]>large)
        {
        large=num[i];
        }
        if(num[i]<small)
        {
                small=num[i];
        }

   }
        printf("Largest number=%d\n",large);
        printf("Smallest number=%d",small);
        return 0;
}
```

Assignment:

Write a C program using array to find largest and smallest number from the list of 100 given numbers.

10) Write a program to input n elements in a 1-D array and check is a given number is present in the array or not. If present display the position of number in array.

```
#include<stdio.h>
int main()
{
int num[100],i,n,search,flag=0,pos;
printf("Enter the number of array elements\n");
scanf("%d",&n);
printf("Enter %d elements\n",n);
for(i=0;i<n;i++)
{
scanf("%d",&num[i]);
}
printf("Enter the  element that you want to  search \n");
scanf("%d",&search);
for(i=0;i<n;i++)
{
          if(search==num[i])
          {
             flag=1;
             pos=i;
             break;
          }
}
if(flag==1)
{
printf("your number is found at index num[%d]",pos)         ;
}
else
{
   printf("your number is not found");
 }
   return 0;
}
```

Assignment:

Write a program to search an element in one-dimensional array containing five integer elements.

11) Write a program to input n number in an array and sort them in ascending order.

```
#include<stdio.h>
int main()
{
  int num[100],i,j,n,temp;
  printf("Enter the number of  array elements\n");
 scanf("%d",&n);
   printf("Enter %d elements\n",n);
   for(i=0;i<n;i++)
    {
      scanf("%d",&num[i]);
    }
     printf("Array before sorting are\n");
     for(i=0;i<n;i++)
      {
      printf("%d\n",num[i]);
     }
    for(i=0;i<n-1;i++)
          {
              for(j=0;j<n-i-1;j++)
            {
                    if(num[j]>num[j+1])
                    {
                    temp=num[j];
                    num[j]=num[j+1];
                    num[j+1]=temp;
                    }
            }
          }
   printf("Sorted array in ascending order are\n");
   for(i=0;i<n;i++)
     {
       printf("%d\n",num[i]);
     }
          return 0;
}
```

12) WAP to input n number in an array and sort them in Descending order.

```
#include<stdio.h>
int main()
{
  int num[100],i,j,n,temp;
  printf("Enter the number of  array elements\n");
 scanf("%d",&n);
   printf("Enter %d elements\n",n);
   for(i=0;i<n;i++)
    {
      scanf("%d",&num[i]);
    }
     printf("Array before sorting are\n");
     for(i=0;i<n;i++)
      {
      printf("%d\n",num[i]);
     }
    for(i=0;i<n-1;i++)
          {
              for(j=0;j<n-i-1;j++)
            {
                    if(num[j]<num[j+1])
                    {
                    temp=num[j];
                    num[j]=num[j+1];
                    num[j+1]=temp;
                    }
     }
}
   printf("Sorted array in descending order are\n");
   for(i=0;i<n;i++)
     {
       printf("%d\n",num[i]);
     }
          return 0;
}
```

13) Write a program to read n numbers and find third largest element among n numbers.

[Hint: Read n numbers and stored them in array and sort them in descending order and print the value of third index i.e num[2])

```
#include<stdio.h>
int main()
{
  int num[100],i,j,n,temp;
  printf("Enter number of  array elements\n");
 scanf("%d",&n);
   printf("Enter %d elements\n",n);
   for(i=0;i<n;i++)
    {
      scanf("%d",&num[i]);
    }
    for(i=0;i<n-1;i++)
          {
              for(j=0;j<n-i-1;j++)
              {
                    if(num[j]<num[j+1])
                    {
                    temp=num[j];
                    num[j]=num[j+1];
                    num[j+1]=temp;
                    }
              }
          }
   printf("Third largest element=%d",num[2]);
  return 0;
}
```

14) Write a program to read marks of n students and print the marks of top five.

```
#include<stdio.h>
int main()
{
	float marks[100],temp;
	int i,j,n;
	printf("Enter the number of students\n");
	scanf("%d",&n);
	printf("Enter the marks of  %d students\n",n);
	for(i=0;i<n;i++)
	{
	scanf("%f",&marks[i]);
	}
	printf("Entered marks of student are\n");
	for(i=0;i<n;i++)
	{
	printf("%f\n",marks[i]);
	}
	for(i=0;i<n-1;i++)
	{
	  for(j=0;j<n-i-1;j++)
	  {
		if(marks[j]<marks[j+1])
		{
		temp=marks[j];
		marks[j]=marks[j+1];
		marks[j+1]=temp;
		}
	  }
	}
	printf("Marks of Top five students are:\n");
	for(i=0;i<5;i++)
	{
	printf("%f\n",marks[i]);
	}
	return 0;
}
```

15) WAP to input n elements in array and copy to another array.

```
#include<stdio.h>
int main()
{
  int a[100],b[100],i,n;
  printf("Enter number of array elements\n");
  scanf("%d",&n);
  printf("Enter %d elements\n",n);
  for(i=0;i<n;i++)
   {
    scanf("%d",&a[i]);
   }
 printf("The entered elements are\n");
 for(i=0;i<n;i++)
  {
   printf("%d\n",a[i]);
  }
  for(i=0;i<n;i++)
  {
  b[i]=a[i];
  }
  printf("\nThe copied elements are\n");
   for(i=0;i<n;i++)
   {
     printf("%d\n",b[i]);
   }
     return 0;
}
```

16) WAP to copy the content of one array to another array in a reverse order.

```
#include<stdio.h>
int main()
{
   int a[100],b[100],i,j,n;
   printf("Enter the number of  array elements\n");
   scanf("%d",&n);
   printf("Enter %d elements\n",n);
   for(i=0;i<n;i++)
   {
     scanf("%d",&a[i]);
    }
    printf("The  array elements are\n");
     for(i=0;i<n;i++)
     {
       printf("%d\n",a[i]);
     }
     for(i=0,j=n-1;i<n;i++,j--)
     {
      b[i]=a[j];
     }
printf("The copied array elements in reverse order
are\n");
     for(i=0;i<n;i++)
     {
      printf("%d\n",b[i]);
     }
     return 0;
  }
```

Two-Dimensional Array (Program solution)

1) **Write a Program to read the matrix of size 3*3 from user and display it to the screen.**

```
#include <stdio.h>
int main()
{
   int arr[3][3];
   int i,j;
  printf("Enter the 9 elements of matrix:\n");
   for(i=0; i<3; i++)
   {
      for(j=0; j<3 ;j++)
      {
         scanf("%d",&arr[i][j]);
      }
   }
   printf("Entered matrix is:\n");
   for(i=0; i<3; i++)
   {
      for(j=0; j<3 ;j++)
      {
         printf("%d\t",arr[i][j]);
      }
      printf("\n");
    }
     return 0;
}
```

Assignment:
*Write a program to read the matrix of size 2*3 from user and display it to the screen***.**

2) **Write a Program to read the matrix of size m*n from user and display it to the screen.**

```
#include<stdio.h>
int main()
{
 int arr[20][20],i,j,m,n;
 printf("Enter the size of row and column of
matrix\n");
 scanf("%d%d",&m,&n);
 printf("Enter %d elements of matrix\n",m*n);
 for(i=0;i<m;i++)
 {
          for(j=0;j<n;j++)
          {
          scanf("%d",&arr[i][j]);
          }
 }
printf("Enterd matrix is\n");
for(i=0;i<m;i++)
{
  for(j=0;j<n;j++)
  {
          printf("%d\t",arr[i][j]);
  }
  printf("\n");
}
return 0;
}
```

3) Write a program to read matrix of order m*n from user and multiply each element of matrix by 3.

```
#include<stdio.h>
int main()
{
 int arr[20][20],i,j,m,n;
 printf("Enter the row size and column size of matrix\n");
 scanf("%d%d",&m,&n);
 printf("Enter %d elements of matrix\n",m*n);
 for(i=0;i<m;i++)
 {
        for(j=0;j<n;j++)
        {
                scanf("%d",&arr[i][j]);
        }
 }
for(i=0;i<m;i++)
 {
        for(j=0;j<n;j++)
        {
                arr[i][j]=3*arr[i][j];
        }
 }
printf("Matrix after multiplication of each element by 3 is\n");
for(i=0;i<m;i++)
{
  for(j=0;j<n;j++)
  {
        printf("%d\t",arr[i][j]);
  }
  printf("\n");
}
return 0;
}
```

4) Write a program to find the sum of all elements of 3 x 3 matrix.

```
#include <stdio.h>
int main()
{
   int arr[3][3],sum=0;
   int i,j;
   printf("Enter the 9 elements of matrix:\n");
   for(i=0; i<3; i++)
   {
      for(j=0; j<3 ;j++)
      {
         scanf("%d",&arr[i][j]);
      }
   }
   for(i=0; i<3; i++)
   {
     for(j=0; j<3 ;j++)
      {
         sum=sum+arr[i][j];
      }
   }
   printf("sum of all elements of matrix = %d",sum);
   return 0;
}
```

Write a program to read values of 3*3 order matrix and find the sum of even elements.

Hint:

```
for(i=0;i<3;i++)
  {
    for(j=0;j<3;j++)
     {
                    if(arr[i][j]%2==0)
                     {
                       esum=esum+arr[i][j];
                     }
     }
  }
```

Write a program to read values of 3*3 order matrix and find the sum of odd elements.

Hint:

```
for(i=0;i<3;i++)
 {
   for(j=0;j<3;j++)
    {
                    if(arr[i][j]%2!=0)
                    {
                      osum=osum+arr[i][j];
                    }
    }
}
```

5) Write a program to input m*n order matrix and find sum of all elements.

```
#include<stdio.h>
int main()
{
 int arr[20][20],i,j,m,n,sum=0;
 printf("Enter the row size and column size of matrix\n");
 scanf("%d%d",&m,&n);
 printf("Enter %d elements of matrix\n",m*n);
 for(i=0;i<m;i++)
 {
        for(j=0;j<n;j++)
        {
                scanf("%d",&arr[i][j]);
        }
 }
for(i=0;i<m;i++)
 {
        for(j=0;j<n;j++)
        {
                sum=sum+arr[i][j];
        }
 }
printf("sum of matrix elements=%d",sum);
return 0;
}
```

Write a program to input m*n order matrix and find sum of all even elements.

Hint:

```
for(i=0;i<m;i++)
 {
        for(j=0;j<n;j++)
        {
                if(arr[i][j]%2==0)
                {
                        esum=esum+arr[i][j];
                }
        }
}
```

Write a program to input m*n order matrix and find sum of all odd elements.

Hint:

```
for(i=0;i<m;i++)
{
        for(j=0;j<n;j++)
        {
                if(arr[i][j]%2!=0)
                {
                        osum=osum+arr[i][j];
           }

        }
}
```

6) Write a program to input m*n order matrix and find sum of each row.

```
#include<stdio.h>
int main()
{
 int arr[20][20],i,j,m,n,sum;
 printf("Enter the row size and column size of matrix\n");
 scanf("%d%d",&m,&n);
 printf("Enter %d elements of matrix\n",m*n);
 for(i=0;i<m;i++)
 {
        for(j=0;j<n;j++)
        {
                scanf("%d",&arr[i][j]);
        }
 }
```

```
for(i=0;i<m;i++)
 {
 sum=0;
 for(j=0;j<n;j++)
  {
         sum=sum+arr[i][j];
  }
  printf("sum of %d row is %d\n",i+1,sum);
}
return 0;
}
```

7) Write a program to input m*n order matrix and find sum of each column.

```
#include<stdio.h>
int main()
{
   int arr[20][20], i, j, m, n, sum;
   printf("Enter the row size and column size of matrix\n");
   scanf("%d%d", &m, &n);
   printf("Enter %d elements of matrix\n", m * n);
   for(i = 0; i < m; i++)
   {
      for(j = 0; j < n; j++)
       {
         scanf("%d", &arr[i][j]);
       }
   }
   for(j = 0; j < n; j++)
    {
      sum = 0;
      for(i = 0; i < m; i++)
     {
         sum = sum+arr[i][j];
      }
      printf("Sum of %d column is %d\n", j + 1, sum);
   }
   return 0;
}
```

8) Write a program to read a matrix and find the sum of all digits in its main diagonal. (diagonal elements from left/trace of matrix)

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
 int arr[20][20],i,j,m,n,sum=0;
 printf("Enter the order of matrix\n");
 scanf("%d%d",&m,&n);
 if(m!=n)
 {
 printf("Invalid order of matrix for this operation");
 exit(0);
 }
 printf("Enter %d elements of matrix\n",m*n);
 for(i=0;i<m;i++)
 {
   for(j=0;j<n;j++)
          {
           scanf("%d",&arr[i][j]);
   }
 }
for(i=0;i<m;i++)
 {
  for(j=0;j<n;j++)
   {
           if(i==j)
            {
              sum=sum+arr[i][j];
            }
   }
 }
 printf("sum of diagonal elements from left=%d",sum)
return 0;
 }
```

Write a Program to read m*n matrix and find the sum of diagonal elements from right.

Hint:

```
if((i+j)==(m-1))
 {
  sum=sum+arr[i][j];
  }
```

9) WAP to input m*n order matrix and find sum of all even and odd numbers and count them.

```
#include<stdio.h>
int main()
{
 int arr[20][20],i,j,m,n,esum=0,osum=0,ecount=0,ocount=0;
 printf("Enter the order of matrix\n");
 scanf("%d%d",&m,&n);
 printf("Enter %d elements of matrix\n",m*n);
 for(i=0;i<m;i++)
 {
          for(j=0;j<n;j++)
          {
           scanf("%d",&arr[i][j]);
          }
 }
  for(i=0;i<m;i++)
  {
    for(j=0;j<n;j++)
     {
                    if(arr[i][j]%2==0)
                     {
                       esum=esum+arr[i][j];
                       ecount++;
                      }
                   else
                    {
                    osum=osum+arr[i][j];
                    ocount++;
                     }
     }
 }
printf("sum of even number=%d\n",esum);
printf("Number even number=%d\n",ecount);
printf("Sum of odd number=%d\n",osum);
printf("Number of odd number=%d",ocount);
return 0;
}
```

10) WAP to input m*n order matrix and find the sum of particular row requested by user.

```
#include<stdio.h>
int main()
{
 int arr[20][20],i,j,row,sum=0,m,n;
 printf("Enter the  order of matrix\n");
 scanf("%d%d",&m,&n);
 printf("Enter %d elements of matrix\n",m*n);
 for(i=0;i<m;i++)
 {
          for(j=0;j<n;j++)
          {
           scanf("%d",&arr[i][j]);
          }
 }
printf("Enter the row number (1 to %d) \n", m);
scanf("%d", &row);
if (row < 1 || row > m)
{
  printf("Invalid row number\n");
}
else
{
  for (j=0;j<n;j++)
  {
  sum =sum+ arr[row-1][j];
  }
  printf("The sum of row %d is: %d\n", row, sum);
}
return 0;
}
```

11) WAP to input m*n matrix and find sum of particular column requested by user.

```
#include<stdio.h>
int main()
{
   int arr[20][20], i, j, column, sum = 0, m, n;
   printf("Enter the order of the matrix \n");
   scanf("%d%d", &m, &n);
   printf("Enter %d elements of the matrix:\n", m * n);
   for (i = 0; i < m; i++)
   {
      for (j = 0; j < n; j++)
      {
         scanf("%d", &arr[i][j]);
      }
   }
   printf("Enter the column number (1 to %d):\n", n);
   scanf("%d", &column);
   if (column < 1 || column > n)
   {
      printf("Invalid column number\n");
   }
   else
   {
      for (i = 0; i < m; i++)
      {
         sum = sum+arr[i][column - 1];
      }
printf("The sum of column %d is: %d\n", column, sum);
   }
   return 0;
}
```

12) WAP to input m*n order matrix and find its transpose.

```
#include<stdio.h>
int main()
{
 int arr[20][20],i,j,m,n;
 printf("Enter the number of rows and column of matrix\n");
 scanf("%d%d",&m,&n);
 printf("Enter %d elements of matrix\n",m*n);
 for(i=0;i<m;i++)
 {
          for(j=0;j<n;j++)
          {
            scanf("%d",&arr[i][j]);
          }
 }
printf("Entered matrix is :\n");
for(i=0;i<m;i++)
{
  for(j=0;j<n;j++)
  {
          printf("%d\t",arr[i][j]);
  }
  printf("\n");
}
printf("Transpose of the  matrix  is \n");
for(i=0;i<n;i++)
{
  for(j=0;j<m;j++)
  {
          printf("%d\t",arr[j][i]);
  }
  printf("\n");
}
return 0;
}
```

13) WAP to find transpose of 4*5 matrix.

```
#include<stdio.h>
int main()
{
 int arr[4][5],i,j;
 printf("Enter 20 elements of matrix\n");
 for(i=0;i<4;i++)
 {
          for(j=0;j<5;j++)
      {
                  scanf("%d",&arr[i][j]);
       }
 }
printf("Enterd matrix is :\n");
for(i=0;i<4;i++)
{
  for(j=0;j<5;j++)
  {
          printf("%d\t",arr[i][j]);
  }
  printf("\n");
}
printf("Transpose of the matrix  is \n");
for(i=0;i<5;i++)
{
  for(j=0;j<4;j++)
  {
          printf("%d\t",arr[j][i]);
  }
  printf("\n");
}
return 0;
}
```

Assignment:
Write a program to find the transpose of 4*3 matrix.

14) WAP to input m*n order matrix and find Largest element.

```
#include<stdio.h>
int main()
{
 int arr[20][20],i,j,m,n,large;
 printf("Enter the order of matrix \n");
 scanf("%d%d",&m,&n);
 printf("Enter %d elements of matrix\n",m*n);
 for(i=0;i<m;i++)
 {
          for(j=0;j<n;j++)
          {
           scanf("%d",&arr[i][j]);
   }
 }
large=arr[0][0];
 for(i=0;i<m;i++)
 {
  for(j=0;j<n;j++)
   {
           if(arr[i][j]>large)
            {
              large=arr[i][j];
            }
   }
 }
printf("largest element=%d",large);
return 0;
}
```

15) WAP to input m*n order matrix and find Smallest element.

```
#include<stdio.h>
int main()
{
 int arr[20][20],i,j,m,n,small;
 printf("Enter the order of matrix\n");
 scanf("%d%d",&m,&n);
 printf("Enter %d elements of matrix\n",m*n);
 for(i=0;i<m;i++)
 {
          for(j=0;j<n;j++)
          {
           scanf("%d",&arr[i][j]);
   }
 }
small=arr[0][0];
 for(i=0;i<m;i++)
 {
  for(j=0;j<n;j++)
   {
           if(arr[i][j]<small)
            {
              small=arr[i][j];
            }
   }
 }
printf("smallest element=%d",small);
return 0;
}
```

16) WAP to input m*n order matrix and convert it to the upper triangular matrix.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
 int arr[20][20],i,j,m,n;
 printf("Enter the order of matrix\n");
 scanf("%d%d",&m,&n);
 if(m!=n)
 {
 printf("Invalid order of matrix for this operation\n");
 exit(1);
}
 printf("Enter %d elements of matrix\n",m*n);
 for(i=0;i<m;i++)
 {
            for(j=0;j<n;j++)
            {
             scanf("%d",&arr[i][j]);
   }
 }
printf("Entered matrix is\n");
for(i=0;i<m;i++)
{
  for(j=0;j<n;j++)
  {
            printf("%d\t",arr[i][j]);
  }
  printf("\n");
}
printf("The upper triangular matrix is\n");
  for(i=0;i<m;i++)
  {
   for(j=0;j<n;j++)
    {
            if(i>j)
            {
             printf("0\t");
            }
            else
            {
             printf("%d\t",arr[i][j]);
            }
    }
            printf("\n");
 }
return 0 ;
}
```

17) **Write a program to add two 3X3 matrix. Display the sum stored in third matrix.**

```
#include<stdio.h>
int main()
{
int a[3][3],b[3][3],sum[3][3],i,j;
printf("Enter  elements of first matrix\n");
for(i=0;i<3;i++)
{
       for(j=0;j<3;j++)
       {
        scanf("%d",&a[i][j]);
        }
}
printf("Enter elements of second matrix\n");
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
     {
       scanf("%d",&b[i][j]);
     }
}
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
      {
        sum[i][j]=a[i][j]+b[i][j];
      }
}
printf("sum of matrix is :\n");
for(i=0;i<3;i++)
 {
  for(j=0;j<3;j++)
  {
   printf("%d\t",sum[i][j]);
  }
  printf("\n");
 }
return 0;
}
```

Assignment:

*Write a Program to read two 3*3 matrix and subtract them*.

Write a program to input m*n order matrix and convert it to the lower triangular matrix.

Hint:
```
if(i<j) {
        printf("0\t");    }
```

18) WAP to find the norm of a matrix.

```
#include<stdio.h>
#include<math.h>
int main()
{
int arr[20][20],i,j,m,n,sum=0;
float norm;
printf("Enter the order of matrix\n");
scanf("%d%d",&m,&n);
printf("Enter %d elements of matrix\n",m*n);
for(i=0;i<m;i++)
{
        for(j=0;j<n;j++)
        {
        scanf("%d",&arr[i][j]);
        }
}
printf("Entered matrix is :\n");
for(i=0;i<m;i++)
{
    for(j=0;j<n;j++)
   {
    printf("%d\t",arr[i][j]);
    }
    printf("\n");
  }
for(i=0;i<m;i++)
{
for(j=0;j<n;j++)
{
sum=sum+pow(arr[i][j],2);
}
}
norm=sqrt((float)sum);
printf("Norm of the matrix=%f",norm);
return 0;
}
```

19) WAP to read two 3*3 matrix and multiply them.

```
#include<stdio.h>
int main()
{
int a[3][3],b[3][3],mul[3][3],i,j,k;
printf("Enter 9 elements of first matrix\n");
for(i=0;i<3;i++)
{
   for(j=0;j<3;j++)
  {
   scanf("%d",&a[i][j]);
   }
}
printf("Enter 9 elements of second matrix\n");
for(i=0;i<3;i++)
{
   for(j=0;j<3;j++)
  {
    scanf("%d",&b[i][j]);
   }
}
for(i=0;i<3;i++)
{
 for(j=0;j<3;j++)
 {
        mul[i][j]=0;
        for(k=0;k<3;k++)
        {
        mul[i][j]=mul[i][j]+a[i][k]*b[k][j];
        }
 }
}
printf("Multiplication of matrix is :\n");
for(i=0;i<3;i++)
{
        for(j=0;j<3;j++)
        {
        printf("%d\t",mul[i][j]);
        }
        printf("\n");
}
return 0;
}
```

20) WAP to read two matrix and multiply them if possible.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
int a[20][20],b[20][20],mul[20][20],i,j,k,r1,c1,r2,c2;
printf("Enter the row and column of first matrix\n");
scanf("%d%d",&r1,&c1);
printf("Enter the row and column of second matrix\n");
scanf("%d%d",&r2,&c2);
if(c1!=r2)
{
printf("Matix multiplication is not possible");
exit(0);
}
printf("Enter %d elements of first matrix\n",r1*c1);
for(i=0;i<r1;i++)
{
for(j=0;j<c1;j++)
{
 scanf("%d",&a[i][j]);
}
}
printf("Enter %d elements of second matrix\n",r2*c2);
for(i=0;i<r2;i++)
  {
     for(j=0;j<c2;j++)
     {
     scanf("%d",&b[i][j]);
     }
  }
for(i=0;i<r1;i++)
{
     for(j=0;j<c2;j++)
     {
                mul[i][j]=0;
                for(k=0;k<c1;k++)
                {
                mul[i][j]=mul[i][j] +a[i][k]*b[k][j];
                }
     }
}
```

```
    printf("Multiplication of matrix is :\n");
    for(i=0;i<r1;i++)
     {
        for(j=0;j<c2;j++)
        {
        printf("%d\t",mul[i][j]);
        }
           printf("\n");
        }
    return 0;
    }
```

21) **WAP to test whether given two matrix are equal or not.**

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
int a[20][20], b[20][20];
int i, j, m1, n1, m2, n2, flag = 1;
printf("Enter the order of first matrix \n");
scanf("%d%d", &m1, &n1);
printf("Enter the order of second matrix \n");
scanf("%d%d", &m2, &n2);
if (m1!=m2||n1!=n2)
{
printf("Two matrix cannot be compared\n");
exit(0)  ;
}
printf("Enter the elements of first matrix \n");
for (i = 0;i<m1;i++)
{
        for (j = 0; j<n1; j++)
        {
        scanf("%d",&a[i][j]);
        }
}
printf("Enter the elements of second matrix \n");
for (i = 0; i < m2; i++)
{
for (j = 0; j <n2; j++)
        {
        scanf("%d",&b[i][j]);
        }
}
```

```
for (i =0;i<m1;i++)
{
for (j =0;j<n1; j++)
        {
                if (a[i][j] != b[i][j])
                {
                        flag = 0;
                        break;
                }
        }
}
        if (flag == 1)
        {
            printf("Two matrices are equal \n");
        }
        else
        {
             printf("Two matrices are not equal \n");
        }
return 0;
}
```

22) Write a program to enter the matrix of size 3X2 and generate new matrix after replacing all even elements by 0.

```
#include<stdio.h>
int main()
{
 int arr[3][2],i,j;
 printf("Enter 6 elements of matrix\n");
 for(i=0;i<3;i++)
 {
        for(j=0;j<2;j++)
   {
   scanf("%d",&arr[i][j]);
   }
 }
printf("Enterd matrix is :\n");
for(i=0;i<3;i++)
{
  for(j=0;j<2;j++)
  {
        printf("%d\t",arr[i][j]);
  }
  printf("\n");
}
```

```
printf("Matrix after replacing all even elements by 0 is\n");
for(i=0;i<3;i++)
{
  for(j=0;j<2;j++)
  {
        if(arr[i][j]%2==0)
        {
              printf("0\t");
        }
        else
        {
              printf("%d\t",arr[i][j]);
        }
}
  printf("\n");
}
return 0;
}
```

23) Write a program that asks a user for a number and find outs if the number is present in 2D array of size m*n.

```
#include<stdio.h>
int main()
{
 int arr[20][20],i,j,m,n,num,flag=0;
 printf("Enter the row size and column size of matrix\n");
 scanf("%d%d",&m,&n);
 printf("Enter %d elements of matrix\n",m*n);
 for(i=0;i<m;i++)
 {
        for(j=0;j<n;j++)
        {
              scanf("%d",&arr[i][j]);
        }
 }
printf("Enter the number that you want to search \n");
scanf("%d",&num);
 for(i=0;i<m;i++)
{
        for(j=0;j<n;j++)
   {
        if(num==arr[i][j])
        {
         flag=1;
         break;
        }
    }
}
```

```
if(flag==1)
{
        printf("your number is found") ;
}
else
{
        printf("your number is not found") ;
}

return 0;
}
```

Strings

Definition

- Strings are sequence of characters stored in consecutive memory location.
- Each character in string occupies one byte of memory.
- Strings always terminated with null character '\0'.

Declaration

```
char string_name[size];
```

Here, size determines the number of characters in string_name.

Eg. char name[5];

Note: *When the compiler assigns a character string to character array, it automatically supplies a null character ('\0') at the end of string. Therefore, size should be equal to maximum number of characters in string plus one.*

Initialization of strings

1. **Compile time initialization**

General form:

char string_name[size]="list of character";

eg. char name[5]="raju";

OR

char string_name[size]={list of character};

eg. char name[5]={'r','a','j','u','\0'};

Note: C also permits to initialize a character array without specifying the number of elements. In such cases size of array will determined automatically,based on number of elements initialized.

Eg. char name[ ]={'r','a','j','u','\0'};

defines, the array string as a five element array.

Program to illustrate compile time initialization of string

```
#include<stdio.h>
int main()
{
        char greeting[6]={'H','e','l','l','o','\0'};
        char name[5]="raju";
        printf("Greeting message:%s\n",greeting);
        printf("Name: %s\n",name);
   return 0;
}
```

Output:

```
Greeting message:Hello
Name:raju
```

2. **Runtime initialization**

The familiar input function scanf() can be used with %s format specification to read string.

Example:

```
char name[20];

scanf("%s",name);
```

Limitation: Here, string variable takes only single word, It is because when whitespace is encountered the scanf() function terminates,

Example:

```
Enter your name: Ramesh Rijal
Your name is: Ramesh
```

To overcome this problem the gets() function is used to read a string of text containing whitespaces, until newline character is encountered.

```
#include<stdio.h>
int main()
{
        char name[20];
        printf("Enter your name:");
        gets(name);
        printf("Your name is:");
        puts(name);
        return 0;
}
```

Output:

```
Enter your name: Dennis Ritchie
Your name is: Dennis Ritchie
```

Array of strings

Array of strings means two dimensional array of characters.

Eg. char name[5][10];

Here, first dimension tells, how many strings can be stored in array. The second dimension tells the maximum length of each string.

In above declaration, we can store 5 strings, each can store maximum 9 characters. Last 10^{th} space is for null terminator in each string.

Program to input name of 5 students and display it.

```
#include<stdio.h>
int main()
{
        int i;
        char name[5][20];
        printf("Enter the name of 5 students\n");
        for(i=0;i<5;i++)
        {
          gets(name[i]);
        }
        printf("The names are\n");
        for(i=0;i<5;i++)
        {
          puts(name[i]);
        }
        return 0;
}
```

String handling function

Write a short note on:
String handling functions.
OR
Explain any seven functions related to string.

String handling function makes the manipulation of string easier and available in string.h header file. Some of them are as follows:

Functions	Purpose	Syntax
strlen()	Finds the length of a string excluding null character	integer_variable=strlen(string);
strcpy()	Copies one string to another including null character	strcpy(destination_string,source_string);
strrev()	Reverses all characters in string except null character	strrev(string);
strcmp()	Compares two strings to find out whether they are same or different. This functions accepts two strings as parameters and returns an integer whose value is i) Less than 0 if the first string is less than second ii) Equal to 0 if both are same iii) Greater than 0 if first string is greater than second	integer_variable=strcmp(string1,string2);
strcat()	Concatenates two strings. i.e. it appends one string at the end of other	strcat(string1,string2);
strlwr()	converts the uppercase string into lowercase	strlwr(string);
strupr()	converts the lowercase string into uppercase	strupr(string);

```
#include<stdio.h>
#include<string.h>
int main ()
{
  int  l ;
  char str1[20] = "pokhara";
  char str2[20] = "university";
  char str3[20];
  l=strlen(str1);
  printf("Length of the str1 is %d\n",l);
  strcpy(str3,str1);
  printf("string copied from str1 to str3 is %s\n",str3 );
  strcat(str1,str2);
  printf("string after concatenation(s1+s2)is %s\n",str1);
  strrev(str3);
  printf("Reverse  of string str3 is %s\n",str3);
  strupr(str3);
  printf("Uppercase of string str3 is %s\n",str3);
  return 0;
}
```

Output:

```
Length of the str1 is 7
string copied from str1 to str3 is pokhara
string after concatenation(s1+s2)is pokharauniversity
Reverse  of string str3 is arahkop
Uppercase of string str3 is ARAHKOP
```

String handling functions

String handling functions makes manipulation of strings easier. String handing functions are available in string.h header file. Some of them are as follows:

a) strlen()

This function returns the an integer which denotes the length of a string passed. Length of a string is number of characters present in it, excluding the terminating null character.

Its syntax is :

integer_variable =strlen(string);

Program

```
#include<stdio.h>
#include<string.h>
int main()
 {
        char str[20];
        int len;
        printf("Enter the string:");
        gets(str);
        len=strlen(str);
        printf("The length of a string is %d",len);
        return 0;
 }
```

Output:

```
Enter the string:ramesh
The length of  a string is 6
```

b) strcpy()

This function copies one string to another. The function accepts two strings as parameters and copies the second string character by character into first one upto including the null character of the second string.
The syntax is :
strcpy(destination_string,source_string);
i.e. strcpy(str2,str1) means the content of str1 is copied to str2.

Program

```
#include<stdio.h>
#include<string.h>
int main()
{
        char str1[20]="ramesh",str2[20];
        strcpy(str2,str1);
        printf("The copied string is %s",str2);
        return 0;
}
```

output:

```
The copied string is ramesh
```

c) strcat()

This function concatenates two strings i.e it appends one string at the end of another. This function accepts two strings as parameters and stores the contents of second string at the end of first. Its syntax is :
strcat (string1,string2);
ie.string1=string1+string2

```
#include<stdio.h>
#include<string.h>
int main()
{
        char str1[20]="pokhara",str2[20]="university";
         strcat(str1,str2);
        printf("The concatenated string is %s",str1);
        return 0;
}
```

Output:

```
The concatenated string is pokharauniversity
```

d) strcmp()

This function compares two strings to find out whether they are same or different. This function accepts two strings as parameters and returns an integer whose value is
i)less than 0 if the first string is less than second
ii)equal to 0 if both are same
iii)greater than 0 if first string is greater than second

The two strings are compared character by character until there is a mismatch or end of one string is reached. Whenever two characters in two string differ, the string which has the character with higher ASCII value is greater.
For example, consider two strings "ram" and "rajesh".The first two characters are same but the third character in string ram and that is in rajesh are different. Since ASCII value of character m in string is ram is greater than that of j in string rajesh, the string ram is greater than rajesh.
Its syntax is:
integer_variable=strcmp(string1,string2);

Program

```
#include<stdio.h>
#include<string.h>
int main()
{
        char str1[20],str2[20];
        int diff;
        printf("Enter the first string:");
        gets(str1);
        printf("Enter the second string:");
        gets(str2);
        diff=strcmp(str1,str2);
         if(diff>0)
        {
        printf("%s is greater than %s",str1,str2);
        }
        else if(diff<0)
        {
         printf("%s is greater than %s ",str2,str1);
        }
        else
          {
         printf("Both strings are same");
        }
        return 0;
}
```

Output

```
Enter first string: ram
Enter second string: rajesh
ram is greater than rajesh
```

e) strrev()

This function is used to reverse all characters in a string except null character at the end of string. The reverse of string “abc” is “cba”.
It’s syntax is strrev(string);
For example:
strrev(str) means it reverses the characters in string s and stores reversed string in str.

```
#include<stdio.h>
#include<string.h>
int main()
{
        char str[20]="hello";
        strrev(str);
        printf("Reversed string is %s",str);
        return 0;
}
```

Output

Reversed string is olleh

f) strupr()

This function converts the lowercase string into uppercase.

```
#include<stdio.h>
#include<string.h>
int main()
{
        char str[20]="hello";
        strupr(str);
         printf("The uppercase of given string is %s",str);
        return 0;
}
```

Output:

The uppercase of given string is HELLO

g) strlwr()

This function converts the uppercase string into lowercase.

```
#include<stdio.h>
#include<string.h>
int main()
{
        char str[20]="HELLO";
        strlwr(str);
         printf("The lowercase of given string is %s",str);
        return 0;
}
```

Output

The lowercase of given string is hello

1) **Write a program to sort n students name in alphabetical order.**

```
#include<stdio.h>
#include<string.h>
int main()
{
   int i,j,n;
   char name[100][20],temp[20];
   printf("Enter the number of students\n");
   scanf("%d",&n);
   printf("Enter the name of %d students:\n",n);
   for(i=0;i<n;i++)
    {
      gets(name[i]);
    }
   for(i=0;i<n-1;i++)
      {
            for(j=0;j<n-1-i;j++)
             {
               if(strcmp(name[j],name[j+1])>0)
                 {
                    strcpy(temp,name[j]);
                    strcpy(name[j],name[j+1]);
                    strcpy(name[j+1],temp);
                  }
                 }
       }
printf("Name of students in alphabetical  order are\n");
    for(i=0;i<n;i++)
     {
      puts(name[i]);
    }
    return 0;
 }
```

Assignment:

Write a program to read n employees names and display them in alphabetical order.

2) **Write a program to insert a given character at a given array index of a given string.For example if the given string is "Gnesh", given character is 'a',and the given array index is 1,the resulting string should be "Ganesh".**

```
#include<stdio.h>
#include<string.h>
int main()
{
 char str[20],ch;
 int len,i,pos;
 printf("Enter the string\n");
 gets(str);
 printf("Enter the index position you want to
insert\n");
 scanf("%d",&pos);
 printf("Enter the character you want to insert\n");
 scanf("%c",&ch);
 len=strlen(str);
 for(i=len+1;i>pos;i--)
 {
          str[i]=str[i-1];
 }
 str[pos]=ch;
 printf("Resulting string is \n");
 puts(str);
 return 0;

}
```

Note: ***This solution is similar for this question.***

Write a program to insert a given character in the array index of string. For example if the string is 'Nepl',given character is 'a', and the given array index is 3,resulting string should be Nepal

3) Write a program to check whether the given string is palindrome or not.(palindrome is a word which reads same from left to right and right to left.eg LIRIL,MADAM etc).

```
#include<stdio.h>
#include<string.h>
int main()
{
 char str1[20],str2[20];
 printf("Enter the string\n");
 gets(str1);
 strcpy(str2,str1);
 strrev(str2);
 if(strcmp(str1,str2)==0)
 {
   printf("String is palindrome");
 }
 else
 {
  printf("String is not palindrome");
 }
 return 0;
}
```

4) Program to check whether a given string is palindrome or not, without using string handling function.

```
#include<stdio.h>
int main()
{
 char str[20];
 int i,len=0,flag=1;
 printf("Enter the string\n");
 gets(str);
 for(i=0;str[i]!='\0';i++)
   {
     len++;
    }
 for(i=0;i<len/2;i++)
 {
          if(str[i]!=str[len-i-1])
           {
                  flag=0;
                  break;
           }
 }
    if(flag==1)
    {
      printf("String is palindrome");
     }
     else
     {
      printf("String is not palindrome");
     }
  return 0;
}
```

5) WAP to convert all the uppercase letter to lowercase and vice versa in a string given by user.

```
#include<stdio.h>
#include<string.h>
#include<ctype.h>
int main()
{
 char str[20];
 int i,l;
 printf("Enter the string\n");
 gets(str);
 l=strlen(str);
  for(i=0;i<l;i++)
  {
          if(islower(str[i]))
           {
                  str[i]=toupper(str[i]);
           }

          else
          {
                  str[i]=tolower(str[i]);
          }

  }
  printf("The converted string is\n");
  puts(str);
  return 0;

}
```

6) WAP to convert all the uppercase letter to lowercase and vice versa in a string given by user.(Alternative solution)

```
#include<stdio.h>
int main()
{
  char str[20];
  int i;
  printf("Enter the string\n");
  gets(str);
  for(i=0;str[i]!='\0';i++)
  {
          if(str[i]>='a'&&str[i]<='z')
           {
                  str[i]=str[i]-32;
           }
          else
          {
                  str[i]=str[i]+32;
          }

  }
  printf("The converted string is %s",str);
  return 0;
}
```

7) **WAP to check whether the Entered character is found in a given string or not.**

```
#include <stdio.h>
#include <string.h>
int main()
{
   char ch, str[50];
   int i,len,flag = 0;
   printf("Enter the string:");
   gets(str);
   printf("Enter character: ");
   scanf("%c", &ch);
   len=strlen(str);
   for (i = 0; i <len; i++)
   {
      if (str[i] == ch)
      {
         flag = 1;
         break;
      }
   }
   if (flag==1)
   {
   printf("character is found in given
string\n");
   }
   else
   {
      printf("Character is not found in given
string\n");
   }
   return 0;
}
```

8) **WAP to input a string and count no of vowels, comma, semicolon, space, digits and constants.**

```
#include<stdio.h>
int main()
{
char str[50];
int i,vowel=0,comma=0,semicolon=0,
space=0,digit=0,consonant=0;
printf("Enter the string\n");
gets(str);
for(i=0;str[i]!='\0';i++)
{
   if(str[i]=='a'||str[i]=='A'||str[i]=='e'||str[i]=='E'
   ||str[i]=='i'||str[i]=='I'||str[i]=='O'||str[i]=='o'
   ||str[i]=='U'||str[i]=='u')
   {
      vowel++;
   }
   else if(str[i]==',')
   {
       comma++;
    }
    else if(str[i]==';')
    {
        semicolon++;
    }
    else if(str[i]==' ')
    {
        space++;
    }
    else if(str[i]>='0'&&str[i]<='9')
    {
        digit++;
     }
     else
     {
        consonant++;
     }
  }
 printf("Number of vowels=%d\n",vowel);
printf("Number of comma =%d\n",comma);
printf("Number of semicolon =%d\n",semicolon);
printf("Number of space =%d\n",space);
printf("Number of digits =%d\n",digit);
printf("Number of consonants=%d",consonant);
return 0;
}
```

9) Write a Program to read string and rewrite in Alphabetical order.

```
#include<stdio.h>
#include<string.h>
int main()
{
char str[20];
int i,j,temp,len;
char name[20];
printf("Enter the string\n");
gets(str);
len=strlen(str);
for(i=0;i<len-1;i++)
{
        for(j=0;j<len-i-1;j++)
        {
                if(str[j]>str[j+1])
                {
                        temp=str[j];
                        str[j]=str[j+1];
                        str[j+1]=temp;
                }
        }
}
puts(str);
return 0;
}
```

10) Write a program to declare name of 10 person in an array in a program already and input any name from the keyboard and check whether it is present in name list or not.

```
#include<stdio.h>
#include<string.h>
int main()
{
  char
list[10][20]={"ram","hari","sita","gopal","narayan","anil",
  "madhav", "manoj","anish","roshan"};
   int i,j,flag=0;
   char name[20];
   printf("Enter the name that you want to search\n");
   gets(name);
   for(i=0;i<10;i++)
    {
        if(strcmp(list[i],name)==0)
          {
            flag=1;
            break;
            }
   }
   if(flag==1)
   {
   printf("Entered name is  found");
}
else
{
 printf("Entered name is not found");
}
    return 0;
}
```

11) WAP to print the following pattern:

```
N
NE
NEP
NEPA
NEPAL
```

```
#include<stdio.h>
#include<string.h>
int main()
{
  char str[6]="NEPAL";
  int i,j,len;
  len=strlen(str);
  for(i=0;i<len;i++)
   {
         for(j=0;j<=i;j++)
         {
          printf("%c",str[j]);
          }
          printf("\n");
    }
    return 0;
}
```

Assignment:
Write a program to print the following pattern.

```
N
EE
PPP
AAAA
LLLLL
```

Hint:
Replace this statement in above program
printf("%c",str[i]);

12) WAP to print the following pattern:

```
1N
2NE
3NEP
4NEPA
5NEPAL
```

```
#include<stdio.h>
#include<string.h>
int main()
{
  char str[6]="NEPAL";
  int i,j,len;
  len=strlen(str);
  for(i=0;i<len;i++)
   {
   printf("%d",i+1);
         for(j=0;j<=i;j++)
         {
          printf("%c",str[j]);
          }
          printf("\n");
    }
    return 0;
}
```

Assignment:
Write a program to print the following pattern.

```
1N
2EE
3PPP
4AAAA
5LLLLL
```

Hint:
Replace this statement in above program
printf("%c",str[i]);

13) Write a program to generate the following pattern by initializing string at first.

```
NEPAL
NEPA
NEP
NE
N
```

```
#include<stdio.h>
#include<string.h>
int main()
{
  char str[6]="NEPAL";
  int i,j,len;
  len=strlen(str);
  for(i=len;i>=0;i--)
   {
         for(j=0;j<i;j++)
         {
          printf("%c",str[j])       ;
           }
          printf("\n");
    }
    return 0;
}
```

Assignment:

Write a program to generate the following pattern by initializing string at first.

PROGRAMMING
PROGRAMMIN
PROGRAMMI
PROGRAMM
PROGRAM
PROGRA
PROGR
PROG
PRO
PR
P

Hint:
Replace this statement in above program
```
  char str[12]="PROGRAMMING";
```

14) What will be the output after executing the following codes.(assume necessary header files)

```
void main()
{
char str[20]={"university"};
char str1[20];
strcpy(str1,"pokhara");
printf("\n%s",str1);
strcat(str1," ");
strcat(str1,str);
printf("\n%s",str1);
strrev(str1);
printf("\n%s",str1);
}
```

Output:

```
pokhara
pokhara university
ytisrevinu arahkop
```

(Trace the program yourself)

15) What will be the output after executing the following codes.(assume necessary header files)

```
void main()
{
        char str[]="HELLO WORLD";
        for(m=0;str[m]!='\0';m++)
        {
                if(m%2==0)
                printf("%c",str[m]);
        }
}
```

Output:

```
HLOWRD
```

(Trace the program yourself)