# ARRAYS

* 1. **Arrays Introduction**
     + A variable is used to store a single value at a time. For example
       - int a;
       - char c;
       - float f;

are of different data types and each can store only a single value of the declared data type. These variables are called scalar variables. A scalar variable is a single variable that can store an atomic value.

* + - An array is a fundamental data structure that enables the storing and manipulation of potentially huge quantities of data.
    - An array stores an ordered sequence of homogeneous values.
    - An array is a collection of individual data elements that is ordered- can count the elements fixed in size.
    - In C, each array has two fundamental properties the data type and size. The individual array elements are identified by an integer index. The array indexing starts from 0. The index is written inside square brackets.
    - Some examples where arrays can be used are

1. List of temperatures recorded every hour in a day, or a month, or a year.
2. List of employees in an organization.
3. List of products and their cost sold by a store.
4. Test scores of a class of students

# One-Dimensional Array

* + - There are several forms of an array like one dimensional and multidimensional.

# Declaration of a One-dimensional Array:

* + - * The array has to be declared in a program before it is used. When defining an array the following need to be specified.
        + the type of data it can hold.
        + the number of values it can hold.
        + name of the array.

***Syntax:***

# data\_type array-name[*size*];

here size specifies the number of elements in the array. The array subscripts can range from 0 to size-1. The lower bound value is 0 and upper bound is size-1.

***Example:***

***int a[10];***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |
| a[0] | a[1] | a[2] | a[3] | a[4] | a[5] | a[6] | a[7] | a[8] | a[9] |

* + - * The array cannot be declared without size in square brackets. ***float f[ ],g[ ];*** *//illegal declaration* **Example:**

#include<stdio.h> #define n 20 main()

{

float x[n]; //is valid

…….

}

Here n is the symbolic constant assigned value of 20.

# Initializing an Array

* + - * As variables are assigned values during declaration arrays can also be initialized in the same manner.

***int a[5]={3,5,7,9,2};***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | 5 | 7 | 9 | 2 |
| a[0] | a[1] | a[2] | a[3] | a[4] |

# Accessing the Array Elements:

* + - * The array elements are accessed by index. If the element 0 and element 3 are to be assigned values then

***a[0]=5; a[3]=8;***

# Example Programs

## Write a C program to initialize array and display them.

#include<stdio.h> main()

{

int a[5]={12,23,34,45,56};

int i; for(i=0;i<5;i++)

{

printf(“%d”,a[i]);

}

}

## Output

12 23 34 45 56

## Write a C program to read array elements from the user and display them

#include<stdio.h> main()

{

int a[5]; int i;

printf(“Enter the elements to be inserted into array”); for(i=0;i<5;i++)

{

scanf(“%d”,&a[i]);

}

printf(“the array elements are as follows”); for(i=0;i<5;i++)

{

printf(“%d”,a[i]);

}

}

## Output

Enter the elements to be inserted into array 10 20 30 40 50 60

the array elements are as follows 10 20 30 40 50 60

# Other Operations on Array

* + - * To increment the ith element the following statements can be used.

***a[i]++;***

***a[i]+=1;***

## a[i]=a[i]+1;

* + - * To add value n to the ith element the following statements may be used

## a[i]+=n; a[i]=a[i]+n;

* + - * To copy the contents of the ith element to kth element.

## a[k]=a[i];

* + - * To copy the contents of one array to another ***int a[10],b[10]; for(i=0;i<10;i++)***

***b[i]=a[i];***

# Example Programs:

## Write a C program to find sum of all elements in an array

#include<stdio.h> main()

{

int a[10],i,n,s=0;

printf(“enter the size of array”); scanf(“%d”,&n);

printf(“enter n elements into array”); for(i=0;i<n;i++)

{

scanf(“%d”,&a[i]);

}

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

{

s=s+a[i];

}

printf(“the sum of elements is %d”,s);

}

## Output

Enter the size 4

Enter 4 elements

23 12 2 6

The sum of elements is 43

## Write a C program to replace even numbers in array by 0.

#include<Stdio.h> main()

{

int a[10],i,n;

printf(“enter the size of array”); scanf(“%d”,&n);

printf(“enter n elements into array”); for(i=0;i<n;i++)

{

scanf(“%d”,&a[i]);

}

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

{

if(a[i]%2==0)

a[i]=0;

}

printf(“the array elements are”); for(i=0;i<n;i++) printf(“%d”,a[i]);

}

## Output

Enter the size of array 6

Enter 6 elements into array 12 23 34 90 45 67

The array elements are

0 23 0 0 45 67

## Write a C program to print the array elements in reverse order.

#include<stdio.h> main()

{

int a[10],i,n;

printf(“enter the size of array\n”); scanf(“%d”,&n);

printf(“enter n elements into array\n”); for(i=0;i<n;i++)

{

scanf(“%d”,&a[i]);

}

printf(“the elements in reverse order”); for(i=n-1;i>=0;i--)

{

printf(“%d\t”,a[i]);

}

}

## Output

Enter the size of array 4

Enter 4 elements into array

12 23 34 45

the elements in reverse order 45 34 23 12

## Write a C program to print the numbers greater than average

#include<stdio.h> main()

{

int a[10],n,i,s=0; float avg;

printf(“enter the number of elements”); scanf(“%d”,&n);

printf(“enter the elements”); for(i=0;i<n;i++)

{

scanf(“%d”,&a[i]); s=s+a[i];

}

avg=(float)s/n;

printf(“the numbers greater than average”); for(i=0;i<n;i++)

{

if(a[i]>avg)

{

printf(“%d”,a[i]);

}

}

}

## Output

Enter the size of array 5

Enter 5 elements

1 4 3 5 2

The numbers greater than average 4 5

## Write a C program to find occurrence of an element in one dimensional array.

#include <stdio.h> int main()

{

int arr[20],n,i; int num,count;

printf("Enter total number of elements: "); scanf("%d",&n);

//read array elements printf("Enter array elements:\n"); for(i=0;i< n;i++)

{

printf("Enter element %d: ",i+1); scanf("%d",&arr[i]);

}

printf("Enter number to find Occurrence: "); scanf("%d",&num);

//count occurance of num count=0;

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

{

if(arr[i]==num) count++;

}

printf("Occurrence of %d is: %d\n",num,count);

}

## Output

Enter total number of elements 5

Enter array elements 12 23 12 45 12

Enter number to find occurrence 12

Occurrence of 12 is 3

## Write a C program to sort given array.

#include<stdio.h> main()

{

int a[10],i,n,j,temp; printf(“enter the size of array”); scanf(“%d”,&n);

printf(“enter n elements into array”); for(i=0;i<n;i++)

{

scanf(“%d”,&a[i]);

}

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

{

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

{

if(a[i]>a[j])

{

temp=a[i]; a[i]=a[j]; a[j]=temp;

}

}

}

printf(“the elements in sorted order”); for(i=0;i<n;i++)

{

printf(“%d”,a[i]);

}

}

## Output:

enter the size of array 5

enter 5 elements into array 4 2 6 1 8

the elements in sorted order

1 2 4 6 8

## Write a C program to find maximum and minimum element in an array

#include<stdio.h> main()

{

int a[10],i,n,max,min; printf(“enter the size of array”); scanf(“%d”,&n);

printf(“enter n elements into array”); for(i=0;i<n;i++)

{

scanf(“%d”,&a[i]);

}

max=a[0]; for(i=0;i<n;i++)

{

if(a[i]>max)

{

max=a[i];

}

}

min=a[0]; for(i=0;i<n;i++)

{

if(a[i]<min)

{

min=a[i];

}

}

printf(“the maximum element is %d”,max); printf(“the minimum element is %d”,min);

}

## Output

enter the size of array 5

enter 5 elements into array 5 7 2 1 9

the maximum element is 9 the minimum element is 1

## Write a C program to search an element in an array using linear search

#include<stdio.h> main()

{

int a[10],i,n,key,f=0; printf(“enter the size of array”); scanf(“%d”,&n);

printf(“enter n elements into array”); for(i=0;i<n;i++)

{

scanf(“%d”,&a[i]);

}

Printf(“Enter the key element to be searched”);

scanf(“%d”,&key); for(i=0;i<n;i++)

{

if(a[i]==key)

{ f=1;

break;

}

}

if(f==1)

printf(“the element found”); else

printf(“element not found”);

}

## Output

Enter the size 5

Enter the elements

12 23 34 45 56

Enter the key element 34

Element found.

## Write a C program to implement binary search

#include<stdio.h> main()

{

int a[10],i,n,l,h,m,key,f=0; printf(“enter the size of array”); scanf(“%d”,&n);

printf(“enter n elements into array”); for(i=0;i<n;i++)

{

scanf(“%d”,&a[i]);

}

printf(“enter the key element”); scanf(“%d”,&key);

l=0;

h=n-1; while(l<=h)

{

m=(l+h)/2; if(a[m]==key)

{ f=1;

break;

}

else if(key<a[m]) h=m-1;

else l=m+1;

}

if(f==1)

{

printf(“element found”);

}

else

{

printf(“element not found”);

}

}

## Output

Enter the size of array 6

Enter 6 elements into array 12 23 34 45 56 67

Enter the key element 56 Element found

## 9.Write a C program to merge two sorted arrays

#include <stdio.h> void main()

{

int array1[5], array2[5], array3[10], m, n, i, j, k = 0; printf("\n Enter sorted elements of array 1: \n");

for (i = 0; i < 5; i++)

{

scanf("%d", &array1[i]);

}

printf("\n Enter sorted elements of array 2: \n"); for (i = 0; i < 5; i++)

{

scanf("%d", &array2[i]);

}

i = 0;

j = 0;

while (i < 5 && j < 5)

{

if (array1[i] < array2[j])

{

array3[k] = array1[i]; i++;

}

else

{

array3[k] = array2[j]; j++;

}

k++;

}

if (i >= 5)

{

while (j < 5)

{

array3[k] = array2[j]; j++;

k++;

}

}

if (j >= n)

{

while (i < m)

{

array3[k] = array1[i]; i++;

k++;

}

}

printf("\n After merging: \n"); for (i = 0; i < 10; i++)

{

printf("\n%d", array3[i]);

}

}

## Output

Enter sorted elements into array1 1 3 5 7 9

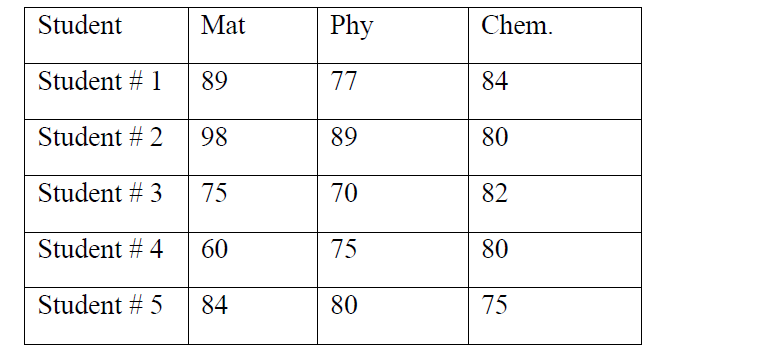
Enter sorted elements into array2 2 4 6 8 10

After merging

1 2 3 4 5 6 7 8 9 10

# Two Dimensional Arrays

* + - 2D arrays are generally known as matrix.
    - There could be situations where a table of values will have to be stored.
    - Consider a student table 3.2 with marks in 3 subjects.

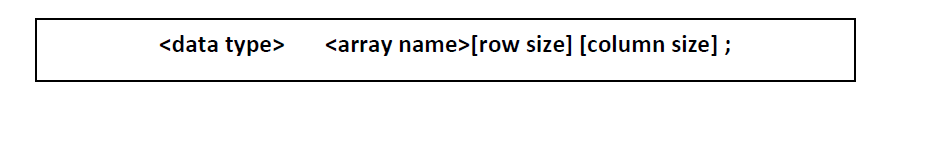


# Table 3.2 Student table

* + - * The above table 3.2 contains a total of 15 values.
      * We can think this table as a matrix consisting of 5 rows & 3 columns.
      * Each row represents marks of student # 1 in all (different) subjects.
      * Each column represents the subject wise marks of all students.
    - In mathematics we represent a particular value in a matrix by using two subscripts such as *Vij.*
    - Here V denotes the entire matrix Vij refers to the value in “ i ”th row and “ j ”th column.
    - Example: In the above table 3.2 V23 refers to the value “80”.
    - C allows us to define such tables of items by using two-dimensional arrays.
    - **Definition**: ***A list of items can be given one variable name using two subscripts and such a variable is called a two – subscripted variable or a two – dimensional array.***

# Two – Dimensional arrays

* + - * They can be declared as.



* + - * The above table 3.2 can be defined in C as

***int V[5][3];***

# Initialization of 2D Array

* + - * There are many ways to initialize two Dimensional arrays –

***int disp[2][4] = {***

***{10, 11, 12, 13},***

## {14, 15, 16, 17}

***};***

# OR

## int disp[2][4] = { 10, 11, 12, 13, 14, 15, 16, 17};

* + - * Things which you must consider while initializing 2D array –
      * When we give values during one dimensional array declaration, we don’t need to mention dimension. But that’s not the case with 2D array; you must specify the second dimension even if you are giving values during the declaration. Let’s understand this with the help of few examples –
        + ***int abc[2][2] = {1, 2, 3 ,4 }*** // Valid declaration
        + ***int abc[][2] = {1, 2, 3 ,4 }*** // Valid declaration
        + ***int abc[][] = {1, 2, 3 ,4 }*** // Invalid declaration – you must specify second dimension
        + ***int abc[2][] = {1, 2, 3 ,4 }*** //Invalid because of the same reason mentioned above

# Store Data into 2D array

## .....

***int abc[5][4];***

## .....

***for(i=0; i<=4; i++)*** //loop for first dimension which is 5 here

***for(j=0;j<=3;j++)*** //loop for 2D of array which is 4 in this example

## {

***printf("Enter value for abc[%d][%d]:", i, j); scanf(“%d”, &abc[i][j]);***

***}***

# Example Programs

## Write a C program to display elements of 2d array

# include<stdio.h> void main( )

{

int i,j;

int a[3][3] = { { 1,2,3}, {4,5,6}, {7,8,9}};

printf(“elements of an array \n \n”); for( i=0; i<3; i++)

{

for ( j=0; j<3; j++)

{

printf (“%d\t”, a[ i ][ j ]);

} // end of inner for loop printf(“\n”);

} // end of outer for loop

} // end of main() function

## Output

Elements of an Array 1 2 3

4 5 6

7 8 9

## Write a C program to read the matrix of order upto 10 x 10 elements and display the same in matrix form.

# include<stdio.h> void main( )

{

int i, j, row, col, a[10][10];

printf(“\n Enter Matrix Order upto (10 x 10) A :”); scanf(“ %d %d ”, &row, &col);

printf(“\n Enter Elements of matrix A: \n”); for( i=0; i<row ; i++)

{

for( j=0; j<col; j++)

{

scanf(“%d”, &a[ i ][ j ]);

}

}

printf(“ \n The matrix is: \n”); for( i=0; i<row; i++)

{

for( j=0; j<col; j++)

{

printf(“ %d ”, a[ i ][ j ]);

}

printf(“\n”);

}

}

## Output

Enter order of matrix upto (10 x 10) A : 3 3 Enter Elements of a matrix A :

3 5 8

4 8 5

8 5 4

The matrix is 3 5 8

4 8 5

8 5 4

## Write a C program to find transpose of given matrix

* + *The transpose of a given matrix is formed by interchanging the rows and columns of a matrix.*

#include <stdio.h> int main()

{

int i,j, row, col, a[10][10], b[10][10];

printf(“ \n Enter order of matrix upto (10 x 10) A:”); scanf(“ %d %d ”, &row, &col);

printf(“\n Enter Elements of matrix A: \n”); for( i=0; i < row; i++)

{

for( j=0; j<col; j++) scanf(“ %d ”, &a[ i ][ j ]);

} /\* transposing logic simply copying one matrix elements to another in reverse order \*/ for( i=0; i < row; i++)

{

for( j=0; j < col; j++) b[ j ][ i ]=a[ i ][ j ];

}

printf(“ \n The Matrix Transpose is \ n”); for( i=0; i<row; i++)

{

for( j=0; j<col; j++) printf(“%d”, b[ i ][ j ]); printf (“ \ n”);

}

}

## Output

Enter order of matrix upto (10 \* 10) A : 3 3 Enter Elements of matrix A:

3 5 8

4 8 5

8 5 4

The Matrix Transpose is

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *3* | *4* | *8* |  | | | | | | | | | |
| *5* | *8* | *5* |  |  |  |  |  |  |  |  |  |  |
| *8* | *5* | *4* |  |  |  |  |  |  |  |  |  |  |
| ***4.*** |  | ***Write*** | ***a*** | ***C*** | ***program*** | ***to*** | ***find*** | ***sum*** | ***and*** | ***difference*** | ***of*** | ***two*** |

## matrices

#include <stdio.h> int main()

{

int i,j,r1,c1, a[10][10], b[10][10];

printf(“Enter Order of Matrix A & B upto 10 x 10”); scanf(“%d %d”, &r1, &c1);

printf(“Enter Elements of Matrix of A: \n”); for( i=0; i < r1; i++)

{

for( j=0; j<c1; j++)

scanf(“ %d ”, &a[ i ][ j ]);

}

printf(“Enter Elements of Matrix of B: \ n”); for( i=0; i < r1; i++)

{

for( j=0; j < c1; j++) scanf(“ %d ”, &b[ i ][ j ]);

}

printf(“\n Matrix Addition \n”); for( i=0; i < r1; i++)

{

for( j=0; j < c1; j++)

printf(“%d\t”, a[ i ][ j ] + b[ i ][ j ]); printf (“ \n”);

}

printf(“n Matrix Subtraction/Difference \n”); for( i=0; i < r1; i++)

{

for( j=0; j < c1; j++)

printf(“%d\t”, a[ i ][ j ] – b[ i ][ j ]); printf(“\n”);

}

}

## Output

Enter order of Matrix A & B upto 10 x 10 : 3 3 Enter Elements of Matrix of A:

4 5 8

2 9 8

2 9 4

Enter Elements of Matrix of B: 1 3 5

0 5 4

6 7 2

Matrix Addition

|  |  |
| --- | --- |
| *5* | *8 13* |
| *2* | *14 12* |
| *8* | *16 6* |

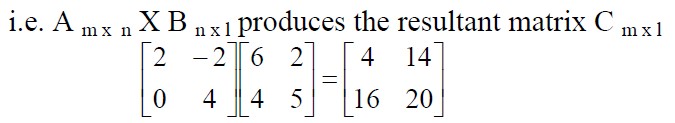
Matrix Subtraction 3 2 3

2 4 4

-4 2 2

## Write a C program to find product of two matrices

* + *Consider two matrices A and B of order 2 x 2. Multiply them to get the resultant matrix C.*



# include<stdio.h> # include<math.h> int main( )

{

int a[10][10], b[10][10], c[10][10], m , n , i , j , l , k ;

printf(“ \n Enter Order of A matrix :”);

scanf(“%d %d”, &m, &n); /\* loop to read values of A matrix \*/ printf(“Enter a matrix \n”);

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

scanf(“%d”, &a[ i ][ j ]);

printf(“\n Enter order of B matrix:”);

scanf(„%d %d”, &n, &l); /\* loop to read values of B matrix \*/ printf(“Enter B matrix \n”);

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

scanf(“%d”, &b[ i ][ j ]);

/\* loop to multiply two matrices \*/ for( i=0; i<m; i++)

{

for ( j=0; j<l; j++)

{

c[ i ][ j ] = 0; for( k=0; k < n; k++)

c[ i ][ j ] = c[ i ][ j ] +a[ i ][ k ] \* b[ k ][ j ];

}

}

/\* loop to print resultant matrix \*/ printf(“ \n Resultant matrix is \n”); for( i=0; i<m; i++)

{

for( j=0; j<l; j++) printf(“%6d”, c[ i ][ j ]); printf(“ \n ”);

}

}

**Explanation**: When this program is executed, the user has to first enter the order (min) of A matrix and its values and its values. The innermost loop

## for (k=0; k<n; k++)

***c[ i ][ j ] = c[ i ][ j ] + a[ i ][ k ] \* b[ k ][ j ];***

is used to multiply row elements of A matrix with respective column elements of B matrix and add the result to get an element for C matrix. This is repeated in the outer loops to get the other elements in the resultant matrix.

## Output

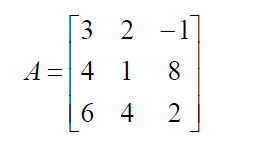
|  |  |  |  |
| --- | --- | --- | --- |
| *Enter order of A* | *matrix :* | *2* | *2* |
| *Enter A matrix* |  |  |  |
| *2 -2* |  |  |  |
| *0 4* |  |  |  |
| *Enter order of B* | *matrix :* | *2* | *2* |
| *Enter B matrix* |  |  |  |
| *6 2* |  |  |  |
| *4 -5* |  |  |  |
| *Resultant matrix* | *is* |  |  |

4 14

16 -20

## Write a C program to find the trace of a given square matrix of order m x m.

* + *We know that the trace of a matrix is defined as the sum of the leading diagonal elements.*
  + *Note that trace is possible only for a square matrix.*



* + *Ex: Trace of A matrix = A11+A22+A33=3+1+2=6*
  + *Row i and column j are equal for a diagonal element. # include<stdio.h>*

main( )

{

int a[10][10], m,i,j, sum;

printf (“\n Enter order of the square matrix :”) ; scanf (“%d”, &m); /\* loop to read values of A matrix \* / printf (“ \n Enter the matrix \n”);

for( i=0; i<m;i++) for ( j=0; j<m; j++)

scanf (“%d”, &a[ i ][ j ]);

/\* loop to find trace of the matrix \* / sum = 0;

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

sum = sum + a[ i ][ i ];

printf (“\n trace of the matrix = %d”, sum);

}

* + *When this program is executed, the user has to enter the order m & values of the given matrix.*
  + *A for loop is written to find the sum of the diagonal elements.*
  + *The index variable of loop i is used for row & column subscripts to represent the diagonal elements.*

Output

Enter order of the square matrix 3 Enter the matrix

3 2 -1

4 1 8

6 4 2

Trace of the matrix = 6

# STRINGS

* 1. **Introduction to Strings**
     + A character array represents a string which ends with Null character (\0).
     + Null character is an escape sequence with ASCII value 0.
     + Null charcter is placed automatically at end of the string.

# Declaration of string

## Syntax:

***char varname[size];***

* + - eg: char A[10];
    - A is a character array of size 10.

# Initialization of string

* + - Like variables , strings are also initialized in 2 ways
    - Static initialization/compile time
    - Dynamic initialization/Run time
    - Static initialization is done in two ways

1. Assigning string to character array
2. Assigning character by character to character array

***Case 1: char s[10]=”Gitam university”;***

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| G | i | t | a | m |  | u | n | i | v | e | r | s | i | t | y | \0 |

S 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

* + - Compiler automatically inserts Null character at the end

## Case 2: char s[10]=

***{'G','i','t','a','m',','u','n','i','v','e','r','s','i','t','y'***

## ,'\0'};

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| G | i | t | a | m |  | u | n | i | v | e | r | s | i | t | y | \0 |

S 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

* + - If data is given character by character '\0' is added manually.

# Example Programs

## Write a program to read and print string using scanf(),printf()

#include<stdio.h> main()

{

char a[10];

printf("enter a string"); scanf("%s",a); printf("given string is"); printf("%s",a);

}

## Output 1

enter a string gitam given string is gitam **Output 2**

enter a string gitam university given string is gitam

## Write a program to read and print sting using gets() and puts()

#include<stdio.h> main()

{

char a[10];

printf("enter a string"); gets(a);

printf("given string is");

puts(a);

}

## Output 1

enter a string hello given string is hello **Output 2**

enter a string hello world given string is hello world

# String handling functions/ string library functions

* + - Functions which are used to perform operations on strings are known as string handling functions
    - These functions are included in **string.h** header file.
    - Some commonly used string functions are
      * strlen() - To find length of the given string
      * strcmp() - To compare given 2 strings
      * strcpy() - To copy one string to another
      * strcat() - To append/ add second string at the end of first string.h
      * strrev() - To reverse the given string
      * strupr() - To convert given lower case string into uppercase
      * strlwr() - To convert given lower case string into uppercase
      * strlen() - This funciton returns length of the string ,which excludes Null character

# strlen(): To find length of the given string

***eg: strlen(“C in depth”); returns length as 10***

# Example Programs

## Write a C program to print string length using string handling functions

#include<stdio.h> main()

{

char s[10]; int x;

printf("enter a string"); gets(s);

x=strlen(s);

printf("length is %d\n",x);

}

## Output

enter a string array concept length is 13

## Write a C program to print string length without using string handling functions

#include<stdio.h> main()

{

char s[10]; int len=0,i;

printf("enter a string\t"); gets(s); for(i=0;s[i]!='\0';i++)

{

len=len+1;

}

printf("lenth of string is %d",len);

}

## Output

enter a string gitam

lenth of string is 5

# strcmp() : This function compares two strings based on ASCII value.

Strcmp() returns value as 0 if both are equal otherwise non zero.it compares character by character and cmparision stops either at end of string is reacherd or corresponding character in the two strings are not euqal.

***strcmp(s1,s2) value is =0 if s1==s2***

***>0 if s1>s2***

***<0 if s1< s2***

# Example Programs

## Write a C program to compare two strings using string handling functions

#include<stdio.h> main()

{

char s[10],s1[10]; int x;

printf("enter a string"); gets(s);

printf("enter string 2"); gets(s1); printf("strings are"); puts(s);

printf("\n"); puts(s1); x=strcmp(s,s1); if(x==0)

printf("strings are equal");

else if(x>0)

printf("string1 is greater than string2"); else

printf("string1 is lesser than string2");

}

## Output

enter a string gitam enter string 2 university strings are gitam university

string1 is lesser than string2

## Write a C program to compare two strings without using string handling functions

#include<stdio.h> int main()

{

char str1[30], str2[30]; int i;

printf("\nEnter two strings :"); gets(str1);

gets(str2); i = 0;

while (str1[i] == str2[i] && str1[i] != '\0') i++;

if (str1[i] > str2[i]) printf("str1 > str2");

else if (str1[i] < str2[i]) printf("str1 < str2"); else

printf("str1 = str2"); return (0);

}

## Output 1

Enter two strings :hello hai

str1 > str2

## Output 2

Enter two strings :abc acc

str1 < str2

## Output 3

Enter two strings :hello hello

str1 = str2

# strcpy() - This function is used to copy string2 data into string1.

## Syntax

***Strcpy(str1,str2);***

# Example programs

## Write a C program to compare two strings using string handling functions

#include<stdio.h> main()

{

char s[10],s1[10]; int x;

printf("enter a string"); gets(s);

printf("enter string 2"); gets(s1); printf("strings are"); puts(s);

printf("\n"); puts(s1);

strcpy(s,s1); printf("string 1 is "); puts(s);

printf("\n string 2 is "); puts(s1);

}

## Output

enter a string gitam enter string 2 university strings are gitam university

string 1 is university string 2 is university

## Write a C program to compare two strings without using string handling functions

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

main()

{

char a[10],b[10]; int i;

printf("enter 2 stirng"); gets(a);

printf("enter 2 stirng"); gets(b);

printf("given string is"); puts(a);

puts(b); for(i=0;b[i]!='\0';i++)

{

a[i]=b[i];

} a[i]='\0'; puts(a);

}

## Output 1

enter 2 stirnggitam

enter 2 stirnguniverstiy iven string isgitam universtiy

after copy universtiy universtiy

## Output 2

enter 2 stirngprogramming wiith c enter 2 stirnggitam

iven string isprogramming wiitgitam gitam

after copy gitam gitam

# strcat( ): This function is used to join 2 Strings.

## Syntax: strcat(s1,s2);

***Eg: string1- happy string2 – hello***

***strcat(string1,string2)- string1-happyhello string2-hello***

# Example Programs

## Write a C program to concatenate two strings using string handling functions

#include<stdio.h> main()

{

char s[10],s1[10]; int x;

printf("enter a string"); gets(s);

printf("enter string 2"); gets(s1);

printf("strings are"); puts(s); printf("\n"); puts(s1);

strcat(s,s1); printf("string 1 is "); puts(s);

printf("\n string 2 is "); puts(s1);

}

## Output

enter a string 1 gitam enter string 2 university strings are gitam university

string 1 is gitamuniversity string 2 is university

## Write a C program to compare two strings without using string handling functions

#include<stdio.h> #include<string.h> main()

{

char a[10],b[10]; int i,n,j;

printf("enter string1"); gets(a);

printf("enter 2nd string"); gets(b);

n=strlen(a);

for(i=n-1,j=0;b[j]!='\0';i++,j++)

{

a[i]=b[j];

} a[i]='\0'; puts(a);

puts(b);

}

## Output

enter string1institute enter 2nd stringtechnology institutetechnology technology

# strrev()- This function is used to reverse the given string

***Syntax: strrev(“gitam”); result- matig***

# Example Programs

## Write a C program to reverse a string using string handling functions .

#include<stdio.h> #include<string.h> main()

{

char a[10],b[10];

printf("enter string:\n"); gets(a);

printf("before reverse string is:"); puts(a);

strrev(a);

printf("after reverse operation string is:"); puts(a);

}

Output

enter string:

hello

before reverse string is: hello

after reverse operation string is: olleh

## Write a C program to reverse the given string without using string handling function strrev()

#include<stdio.h> #include<string.h> main()

{

char a[10],b[10]; int n,i,j;

printf("enter string1"); gets(a);

printf("before reverse string is "); puts(a);

n=strlen(a);

for(i=n-1,j=0;i>=0;i--,j++)

{

b[j]=a[i];

} b[j]='\0';

printf("string after reverse "); printf("%s",b);

}

## Output

enter string1 c in depth

before reverse string is c in depth string after reverse htped ni c

# strupr()- This function is used to change the string from lower case to upper case. If string given is upper case it prints as it is on screen.

***Syntax: strupr(“hello”); Output- HELLO***

# Example Programs

## 1. Write a C program to covert a string to uppercase using

**string handling functions** #include<stdio.h> #include<string.h> main()

{

char a[10],b[10]; int n,i,j;

printf("enter string1"); gets(a);

strupr(a);

printf("string after change into upper case is "); puts(a);

}

## Output

enter string1 madam

string after change into upper case is MADAM

# strlwr()- This function is used to change the string from upper case to lower case

***Syntax: strlwr(“HELLO”); output- hello***

# Example Programs

***1. Write a C program to covert a string to lowercase using***

# string handling functions

#include<stdio.h> #include<string.h> main()

{

char a[10],b[10]; int n,i,j;

printf("enter string:"); gets(a);

strlwr(a);

printf("string after change into lower case is: "); puts(a);

}

## Output

enter string MADAM

string after change into lower case is: madam

# Two dimensional character arrays

* + - A string is an array of characters; so, an array of strings is an array of arrays of characters. Of course, the maximum size is the same for all the strings stored in a two dimensional array.

# Declaration:

## Syntax: char varname[size][size]; Example: char a[5][10];

* + - * here a is character array which stores 5 strings and each string of length 10 characters.

# Initialization of strings:

1. Static initialization

***char a[5][5]={“cse”,”ece”,”eee”,”eie”,”it”};***

1. Dynamic initialization

***char a[5][10];***

# Example Programs

## Write a C program to print 5 strings

#include<stdio.h> #include<string.h> main()

{

char a[5][10]={"ece","cse","eie","eee","it"}; int i;

printf("given strings are\n"); for(i=0;i<5;i++) printf("%s\n",a[i]);

}

## Output

given strings are ece

cse eie eee it

## Write a C program to read and print n strings

#include<stdio.h> main()

{

char a[5][10]; int i,n;

printf("enter no of strings"); scanf("%d",&n);

printf("enter %d strings:\n",n); for(i=0;i<n;i++) scanf("%s",a[i]);

printf("given strings are\n"); for(i=0;i<n;i++)

{

printf("%s\n",a[i]);

}

}

## Output

enter no of strings:5

enter 5 strings:

cello sasi surya chandra charitha

given strings are cello

sasi surya chandra charitha

## Write a C program to sort given strings in ascending order.

#include<stdio.h> #include<string.h> main()

{

char a[5][10],t[10]; int i,n,j;

printf("enter no of strings:\n"); scanf("%d",&n);

printf("enter strings:\n "); for(i=0;i<5;i++) scanf("%s",a[i]); for(i=0;i<n-1;i++)

{

for(j=i+1;j<n;j++)

{

if(strcmp(a[i],a[j])>0)

{

strcpy(t,a[i]);

strcpy(a[i],a[j]);

strcpy(a[j],t);

}

}

}

printf("\n After sorting\n"); for(i=0;i<n;i++) printf("%s\t",a[i]);

}

## Output

enter no of strings: 5

enter strings:

ece eee cse it mech

After sorting

cse ece eee it mech

## Write a C program to search given string from set of strings

#include<stdio.h> #include<string.h> main()

{

char a[5][10],b[10]; int i,n;

printf("enter no of strings:"); scanf("%d",&n);

printf("enter strings:\n"); for(i=0;i<5;i++) scanf("%s",a[i]);

printf("enter a search string:"); scanf("%s",b);

printf("gieven strings are: "); for(i=0;i<5;i++) printf("%s\t",a[i]); printf("\n");

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

{

if(strcmp(a[i],b)==0)

{

printf("string found at %d \n",i); break;

}

}

}

## Output

enter no of strings:5 enter strings:

eee ece cse eie mech

enter a search string:mech

gieven strings are: eee ece cse eie mech string found at 4

## Write a C program to perform all string handling functions.

#include<stdio.h> #include<string.h> main()

{

char a[10],b[10]; int ch,len; printf("enter str1"); gets(a); printf("enter str2"); gets(b);

while(1)

{

printf(“\n choose ur option”);

printf("\n1.length\n 2.compare\n 3.copy\n 4.concat\n 5.rev\n6.exit\n");

printf("enter ur choice"); scanf("%d",&ch); switch(ch)

{

case 1: len=strlen(a); printf("length is %d\n",len); break;

case 2:if(strcmp(a,b)==0)

{

printf("both strings are equal\n");

}

else if(strcmp(a,b)>0)

printf("%s is greater than %s\n",a,b); else

printf("%s is greater than %s\n",b,a); break;

case 3: printf(" str1 %s\n",a); printf("str2 %s\n",b); strcpy(a,b);

printf("aafter copy strings are\n"); printf(" str1 %s\n",a);

printf("str2 %s\n",b); break;

case 4:printf(" str1 %s\n",a); printf("str2 %s\n",b);

strcat(a,b);

printf(" str1 %s\n",a); printf("str2 %s\n",b); break;

case 5:strrev(a); printf("%s\n",a); break;

case 6:exit(0);

}

}

}

## Output

enter str1 Hello enter str2 World choose ur option 1.length 2.compare 3.copy

4.concat 5.rev 6.exit 1

Length is 5