#### UNIT II ARRAYS AND STRINGS

Arrays – Initialization – Declaration – One dimensional and two-dimensional arrays String- String operations – String Arrays-Simple programs- sorting- searching – matrix operations.

**Arrays:** An array is a collection of similar data items that are stored under a common name. In other words an array is defined as **finite ordered collection of homogenous** data, stored in contiguous memory locations.

* **finite** *means* data range must be defined.
* **ordered** *means* data must be stored in continuous memory addresses.
* **homogenous** *means* data must be of similar data type.

##### Example where arrays are used,

* to store list of Employee or Student names
* to store marks of a students
* to store list of numbers or characters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | e | i | o | u |

[0] [1] [2] [3] [4] A1: Character Array

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | [0] |  | [1] | [2] |  |  | [3] |  | [4] | [5] |  | A2: Integer array |
| 1.1 |  | 1.2 | 1.3 |  |  | 1.4 | 1.5 | | 1.6 |  |  |
| [0] |  | [1] | [2] |  |  | [3] |  | [4] | [5] |  | A3: Float array |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | [0] |  |  | [1] |  |  | [2] |  |  | [3] |  | A4: User defined array |

#### Characteristics of an array:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | 1 |  |  | B | 2 |  |  | C | 3 |  |  | D | 4 |  |
|  |  |  |  |  |  |  |  |  |  |

* The individual element of an array is not named. They share a common name.
* The individual element of an array is referred or accessed by position in an array.
* Position of an array is specified with an integer value known as index or subscript.
* Array index starts with 0 and ends with n-1.
* The memory size required by an array can be computed as

Array size = (size of element type\*Number of elements in an array). In the above example A1= 1\*5=5, A2=2\*6=12

* An array elements are always stored in continues memory locations.
* The size of an array can’t be expanded at runtime.

##### Types of an array:

1. Single dimension array
2. Multi dimension array

#### Single dimension array

* An array with one subscript is termed as one dimensional array.
* Single dimension or 1D array consists of a fixed no of elements of same data types organized as a simple linear sequence.
* The elements of 1D array is also called as linear array or list.

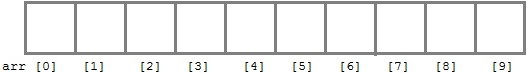
## Declaration of 1D array:

**Example:**

int arr[10];

**Syntax:**

**data type** array\_name[size];



#### Rules for 1D array:

1. Array size can be an expression and a qualified constant.

**Ex**: int array[3+2]; int array2[size];

1. Array size can’t be a float value.

#### Initialization of an Array

* After an 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 at **runtime**.

#### Compile time Array initialization

Compile time initialization of array elements is same as ordinary variable initialization.

**Syntax:**

*data type* **array-name[size] = { list of values };**

int marks[4]={ 67, 87, 56, 77 }; //integer array initialization

float area[5]={ 23.4, 6.8, 5.5 }; //float array initialization

int marks[4]={ 67, 87, 56, 77, 59 }; //Compile time error

|  |  |
| --- | --- |
| **Example:**  #include< stdio.h> #include< conio.h> void main()  {  int i;  int a[]={2,3,4}; //Compile time array initialization for(i=0 ; i<3 ; i++)  printf("%d\n",arr[i]); getch();  } | **Output**  2  3  4 |

#### Runtime Array initialization

An array can also be initialized at runtime using scanf() function.

This approach is usually used for initializing large array, or to initialize array with user specified values.

**syntax:**

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

{

scanf(“%d”,&a[i]);

}

|  |  |
| --- | --- |
| **Example:**  #include< stdio.h> #include< conio.h> void main()  {  int a[50];  int i,j,n;  printf(“Enter the size of an array”); scanf(“%d”,&n);  printf("Enter array elements"); for(i=0;i<n;i++)  {  scanf("%d",&arr[i]); //Run time array initialization  }  printf(“The elements of ur array is”);  for(j=0;j<n;j++)  {  printf("%d\n",arr[j]);  }  getch();  } | ***Output:***  Enter the size of an array 4  Enter array elements 1  2  4  5  The elements of ur array is 1  2  4  5 |

#### Operations on 1D array:

##### Subscripting an 1D array

It is an action of selecting an element from an array.

printf(“%d”, a[5]);

##### Assigning an array to another array

A variable can be assigned to another variable but array can’t be assigned to

another array directly.

|  |  |
| --- | --- |
| #include<stdio.h> | **Output:** |
| #include<conio.h> | //compilation error |
| void main() |  |
| { |  |
| int a[3], b[3]={1,2,3}; |  |
| a[3]=b[3]; |  |
| printf(“%d”, a[0]); |  |
| } |  |

##### How to do:

* + The name of an array refers to the address of the first element of an array, it is a unique constant.
  + One address can’t be assigned to another. so it shows compilation error.
  + It can be done using for loop b assigning individual values to another array.

##### Equating an array with another array

|  |  |
| --- | --- |
| #include<stdio.h> | ***Output:*** |
| #include<conio.h> | Arrays are not equal |
| void main() |  |
| { |  |
| int a[3]=[1,2,3}, b[3]={1,2,3}; |  |
| if (a==b) |  |
| printf(“Arrays are equal”); |  |
| else |  |
| printf(“Arrays are not equal”); |  |
| } |  |

***Programs on one dimensional array:***

1. Program to get and print an array
2. Program to print an array in a reverse order
3. Sum of array elements
4. Find odd or even number of an array
5. Remove the redundant element of an array
6. Delete a particular element of an array.

|  |  |
| --- | --- |
| ***Program to get and print an array*** | ***Output:*** |
|  | Enter no of elements |
| #include<stdio.h> | 4 |
| #include<conio.h> | Enter the values |
| void main() | 80 |
| { | 60 |
| int i, a[50], n; | 50 |
| clrscr(); | 70 |
| printf("Enter the size of an array :"); | The values that you entered |
| scanf("%d", &n); | are |
| printf("Enter the array elemets :"); | 80 |
| for (i = 0; i < n; i++) | 60 |
| { | 50 |
| scanf("\n%d", &a[i]); | 70 |
| } |  |
| printf("The array elements are"); |  |
| for (i = 0; i < n; i++) |  |
| { |  |
| printf("\n%d", a[i]); |  |
| } |  |
| getch(); |  |
| } |  |
| ***Program to print an array in a reverse order*** | ***Output:*** |
| Enter no of elements |
| #include<stdio.h> | 4 |
| #include<conio.h> | Enter the values |
| void main() | 80 |
| { | 60 |
| int i, a[50], n; | 50 |
| clrscr(); | 70 |
| printf("Enter no of elements :"); | The values that you entered |
| scanf("%d", &n); | are |
| printf("Enter the values :"); | 70 |
| for (i = 0; i < n; i++) | 50 |
| { | 60 |
| scanf("\n%d", &a[i]); | 80 |
| } |  |
| printf("The array in reversed order is\n"); |  |
| for (i = n-1; i>=0; i--) |  |
| { |  |
| printf("%d\n", a[i]); |  |
| } |  |
| getch(); |  |
| } |  |

|  |  |
| --- | --- |
| ***Sum of array elements*** | ***Output:*** |
| #include<stdio.h> | Enter no of elements |
| #include<conio.h> | 3 |
| void main() | Enter the values |
| { | 2 |
| int i, a[50], n, sum=0; | 3 |
| clrscr(); | 1 |
| printf("Enter no of elements :"); | the sum of the array is 6 |
| scanf("%d", &n); |  |
| printf("Enter the values :"); |  |
| for (i = 0; i < n; i++) |  |
| { |  |
| scanf("\n%d", &a[i]); |  |
| } |  |
| for (i = 0; i < n; i++) |  |
| { |  |
| sum=sum+a[i]; |  |
| } |  |
| printf("the sum of the array is %d",sum); |  |
| getch(); |  |
| } |  |
| ***Program to find odd or even number of an array*** |  |
| #include<stdio.h> | ***Output:*** |
| #include<conio.h> | Enter no of elements : |
| void main() | 4 |
| { | Enter the values: |
| int i, a[50], n; | 20 |
| clrscr(); | 5 |
| printf("Enter no of elements :"); | 7 |
| scanf("%d", &n); | 14 |
| printf("Enter the values :"); |  |
| for (i = 0; i < n; i++) | 20 is even number |
| { | 5 is odd number |
| scanf("\n%d", &a[i]); | 7 is odd number |
| } | 14 is even number |
| for (i = 0; i < n; i++) |  |
| { |  |
| if(a[i]%2==0) |  |
| printf("%d is even number", a[i]); |  |
| else |  |
| printf(“%d is odd number”,a[i]); |  |
| } |  |
| getch(); |  |
| } |  |

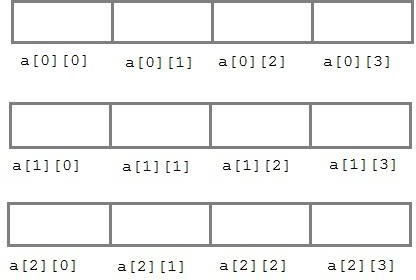
|  |  |
| --- | --- |
| ***Delete a duplicate element in an array*** | ***OUTPUT:*** |
|  | Enter number of elements in |
| #include<stdio.h> | array |
| #include<conio.h> | 5 |
| void main() | Enter the array elements |
| { | 1 |
| int n, a[10],i, j,k; | 5 |
| clrscr(); | 7 |
| printf("Enter number of elements in array\n"); | 8 |
| scanf("%d",&n); | 8 |
| printf("Enter the array elements\n"); | The Resultant array |
| for(i=0;i<n;i++) | 1 |
| scanf("%d",&a[i]); | 5 |
| for(i=0;i<n;i++) | 7 |
| { | 8 |
| for(j=i+1;j<n;j++) |  |
| { |  |
| if(a[i]==a[j]) |  |
| { |  |
| for(k=j;k<n;k++) |  |
| a[k]=a[k+1]; |  |
| n--; |  |
| } |  |
| } |  |
| } |  |
| printf("The Resultant array\n"); |  |
| for(i=0;i<n;i++) |  |
| printf("%d\n",a[i]); |  |
| getch(); |  |
| } |  |
| ***Delete a particular element of an array*** | ***OUTPUT:***  Enter number of elements in array  5  Enter the array elements 1  5  7  8  9  Enter the position to be deleted from an array 1 |
| #include<stdio.h> |
| #include<conio.h> |
| void main() |
| { |
| int n, a[10],i,pos; |
| clrscr(); |
| printf("Enter number of elements in array\n"); |
| scanf("%d",&n); |
| printf("Enter the array elements\n"); |
| for(i=0;i<n;i++) |
| scanf("%d",&a[i]); |
| printf("Enter the position to be deleted"); |
| scanf("%d",&pos); |

|  |  |
| --- | --- |
| if(pos>=n) | The Resultant array |
| printf("deletion is not possible"); | 1 |
| else | 7 |
| { | 8 |
| for(i=pos;i<n-1;i++) | 9 |
| a[i]=a[i+1]; |  |
| printf("The Resultant array\n"); |  |
| for(i=0;i<n-1;i++) |  |
| printf("%d\n",a[i]); |  |
| } |  |
| getch(); |  |
| } |  |

***Two dimensional Arrays***

* + An array with two subscripts is termed as two dimensional arrays.
  + A two dimensional array can be taught of as a group of one or more dimensional arrays, All of which share a common name and distinguished by subscript values.
  + A two dimensional array enables us to store multiple rows of elements, that is a table of values or a matrix.

**Declaration of two dimensional Arrays**



***Syntax:***

data\_type array\_name[row-size][column-size];

**Example :**

int a[3][4];

|  |  |
| --- | --- |
| **Compile-time initialization of two dimensional Array** | **Run-time initialization of two dimensional Arrays** |
| The two dimensional array can be initialized at the time of declaration is known as compile time initialization.  **Example:**  int a[2][3] = { {0,0,0}, {1,1,1} };  int a[2][3] = { 0,0,0, 1,1,1 }; | **syntax:**  for(i=0;i<n;i++)  {  for(j=0;j<n;j++)  {  scanf(“%d”,a[i][j]);  }  } |

### Programs on two dimensional arrays:

1. Get and print 2D array or matrix
2. Matrix addition
3. Matrix subtraction
4. Matrix multiplication
5. Transpose of a matrix
6. Sum of diagonal element of an array

|  |  |
| --- | --- |
| **Get and print 2D array or matrix**  #include< stdio.h> #include< conio.h> void main()  {  int a[3][4]; int i,j,k;  printf("Enter a matrix"); for(i=0;i<2;i++)  {  for(j=0;j<2;j++)  {  scanf("%d",&a[i][j]);  }  printf(“The matrix you entered is”);  }  for(i=0;i<2;i++)  {  for(j=0;j<2;j++)  {  printf("%d\t",a[i][j]);  }  Printf(“\n”);  }  getch();} | **Output:**  Enter a matrix 1  2  3  4  The matrix you entered is 1 2  3 4 |

|  |  |
| --- | --- |
| ***Matrix addition*** #include<stdio.h> #include<conio.h> void main()  {  int i, j, a[2][2], b[2][2], c[2][2];  clrscr();  printf("Enter A matrix\n"); for (i = 0; i < 2; i++)  {  for (j = 0; j < 2; j++)  {  scanf("%d", &a[i][j]);  }  }  printf("Enter B matrix\n"); for (i = 0; i < 2; i++)  {  for (j = 0 ; j < 2; j++)  {  scanf("%d", &b[i][j]);  }  }  printf("Sum of two matrices:-\n"); for (i = 0; i < 2; i++)  {  for (j = 0 ; j < 2; j++)  {  c[i][j] = a[i][j] + b[i][j];  printf("%d\t", c[i][j]);  }  printf("\n");  }  getch();  } | ***OUTPUT:***  Enter A matrix 1  1  1  1  Enter B matrix 2  2  2  2  Sum of two matrices:- 3 3  3 3 |
| ***Matrix subtraction*** #include<stdio.h> #include<conio.h> void main()  {  int i, j, a[2][2], b[2][2], c[2][2];  clrscr();  printf("Enter A matrix\n"); for (i = 0; i < 2; i++)  {  for (j = 0; j < 2; j++) |  |

|  |  |
| --- | --- |
| {  scanf("%d", &a[i][j]);  }  }  printf("Enter B matrix\n"); for (i = 0; i < 2; i++)  {  for (j = 0 ; j < 2; j++)  {  scanf("%d", &b[i][j]);  }  }  printf("Sum of two matrices:-\n"); for (i = 0; i < 2; i++)  {  for (j = 0 ; j < 2; j++)  {  c[i][j] = a[i][j] - b[i][j];  printf("%d\t", c[i][j]);  }  printf("\n");  }  getch();  } | ***OUTPUT:***  Enter A matrix 1  1  1  1  EnterB matrix 2  2  2  2  Subtraction of two matrices:-  -1 -1  -1 -1 |
| ***Matrix multiplication:*** #include<stdio.h> #include<conio.h> void main()  {  int i, j, k;  int a[2][2], b[2][2], c[2][2]={0};  clrscr();  printf("Enter A matrix\n"); for (i = 0; i < 2; i++)  {  for (j = 0; j < 2; j++)  {  scanf("%d", &a[i][j]);  }  }  printf("Enter B matrix\n"); for (i = 0; i < 2; i++)  {  for (j = 0; j < 2; j++)  { | ***OUTPUT:***  Enter A matrix 1  1  1  1  EnterB matrix 2  2  2  2  Product of two matrices:- 4 4  4 4 |

|  |  |
| --- | --- |
| scanf("%d", &b[i][j]);  }  }  printf("The product of two matrix is\n"); for (i = 0; i < 2; i++)  {  for (j = 0; j < 2; j++)  {  for (k = 0; k < 2; k++)  {  c[i][j] = c[i][j] + a[i][k]\*b[k][j];  }  printf("%d\t", c[i][j]);  }  printf("\n");  }  getch();  } |  |
| ***Transpose of a matrix*** | ***OUTPUT:*** |
| #include <stdio.h> | Enter the elements of matrix |
| #include<conio.h> | 1 2 |
| void main() | 3 4 |
| { | Transpose of entered matrix:- |
| int i, j, a[2][2]; | 1 3 |
| clrscr();  printf("Enter the elements of matrix\n"); | 2 4 |
| for (i = 0; i < 2; i++) |  |
| { |  |
| for(j = 0; j < 2; j++) |  |
| { |  |
| scanf("%d",&a[i][j]); |  |
| } |  |
| } |  |
| printf("Transpose of entered matrix :-\n"); |  |
| for (i = 0; i < 2; i++) |  |
| { |  |
| printf("\n"); |  |
| for( j = 0 ; j < 2 ; j++ ) |  |
| { |  |
| printf("%d\t",a[j][i]); |  |
| } |  |
| } |  |
| getch(); |  |
| } |  |

|  |  |
| --- | --- |
| ***Sum of diagonal element of an array*** | ***OUTPUT:*** |
| #include<stdio.h> | Enter a matrix |
| #include<conio.h> | 1 2 |
| void main() | 3 4 |
| { | Sum of the diagonal elements |
| int a[2][2],i,j,sum=0,r,c; | of a matrix is: 5 |
| clrscr(); |
|  |
| printf("\nEnter a matrix: "); |  |
| for(i=0;i<2;i++) |  |
| { |  |
| for(j=0;j<2;j++) |  |
| { |  |
| scanf("%d",&a[i][j]); |  |
| } |  |
| } |  |
| for(i=0;i<2;i++) |  |
| { |  |
| for(j=0;j<2;j++) |  |
| { |  |
| if(i==j) |  |
| sum=sum+a[i][j]; |  |
| } |  |
| } |  |
| printf("\n sum of diagonal element is %d”,sum); |  |
| getch(); |  |
| } |  |

### Multi dimensional array:

* + An array with more than one dimension is generally termed as multi dimension array.
  + They are two dimensional array, three dimensional array and four dimensional

array…… etc.

#### Declaration of three dimensional array:

**Example:**

int a [2][2][2];

**Syntax:**

dataype array\_name[size 1][size 2][size 3];

**Initialization of three dimensional arrays**:

**Syntax:**

dataype array\_name[size 1][size 2][size 3]={list of values};

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

#### Disadvantages of an array:

* + The elements in an array must be same data type.
  + The size of an array is fixed. we can’t expand the size in run time.
  + The insertion and deletion operations in an array require shifting of elements which takes more time.

### Strings

* + **String** is a sequence of characters that is treated as a single data item and terminated by null character '\0'. Remember that C language does not support strings as a data type.
  + A **string** is actually one-dimensional array of characters in C language. These are often used to create meaningful and readable programs.

**For example :** The string "hello world" contains 12 characters including '\0' character which is automatically added by the compiler at the end of the string.

#### Declaring and Initializing a string variables

There are different ways to initialize a character array variable.

1. char name[10]="StudyTonight"; //valid character array initialization 2. char name[10]={'L','e','s','s','o','n','s','\0'}; //valid initialization

Remember that when you initialize a character array by listings all its characters separately then you must supply the '\0' character explicitly.

##### Some examples of illegal initialization of character array are,

char ch[3]="hell"; //Illegal char str[4];

str="hell"; //Illegal

#### String Input and Output

* Input function scanf() can be used with **%s** format specifier to read a string input from the terminal.
* But there is one problem with **scanf()** function, it terminates its input on first white space it encounters. Therefore if you try to read an input string "Hello World" using **scanf()** function, it will only read **Hello** and terminate after encountering white spaces.
* However, C supports a format specification known as the **edit set conversion code %[..]** that can be used to read a line containing a variety of characters, including white spaces.

##### Another method to read character string with white spaces from terminal is gets() function.

char a[20]; gets(a); printf("%s",a);

#include< stdio.h> #include< conio.h> void main()

{

char a[20]; clrscr();

printf("Enter a string"); scanf("%[^\n]",&a);

printf("%s",a); getch();

}

## String Handling Functions

|  |  |  |
| --- | --- | --- |
| **Function** | **Purpose** | **Syntax** |
| strlen | Calculates the length of the string s. | strlen(str1); |
| strcmp | Compares two strings. | strcmp(str1, str2); |
| strcmpi | Compares two strings without case sensitive. | Strcmpi(str1,str2); |
| strcat | Appends the copy of the string *src* to the end of the string *dest*. | strcat(str1, str2); |
| strcpy | Copies the source string *src* to the destination string  *dest*. | strcpy(str1,str2); |
| strrev | Reverse the content of the string *s*. | strrev(str1); |
| strlwr | Converts the string to lowercase. | strlwr(str1); |
| strupr | Converts the string to uppercase. | strupr(str1); |
| strset | Set all character in a string s to the character *ch*. | strset(str1,ch); |
| Strnset | Set the first n character in a string s to the character *ch*. | Strnset(str1,ch,n); |
| strchr | Scans a string for the first occurrence of the given character. | strchr(str1,ch); |
| strrchr | Scans a string for the last occurrence of the given character. | strrchr(str1,ch); |
| strstr | Scans a string for the first occurrence of the substring in another string. | strstr(str1,ch); |

|  |  |  |
| --- | --- | --- |
| **S.no** | **Result type** | **Function** |
| 1 | integer | strlen, strcmp, strcmpi |
| 2 | string | strcat, strcpy, strrev, strupr, strlwr, strset, strnset |
| 3 | address | strchr, strnchr, strstr |

|  |  |
| --- | --- |
| **Programs using predefined fun** | **Programs without using predefined fun** |
| **To find the length of the string** | |
| #include <stdio.h> #include <string.h> #include<conio.h> void main()  {  char a[50]; int l; clrscr();  printf("Enter a string\n");  scanf(“%s”,a); l = strlen(a);  printf("Length of a string is %d\n",l); getch();  } | #include <stdio.h> #include<conio.h> void main()  {  char a[50]; int i, l = 0; clrscr();  printf("Enter a string \n");  scanf(“%s”,a);  for (i = 0; a[i] != '\0'; i++) l++;  printf("Length of a string is %d\n",l); getch();  } |
| **Output:**  Enter a string  **Hai**  Length of a string is **3** | |
| **To compare two strings** | |
| #include <stdio.h> #include<conio.h> #include <string.h> void main()  {  char a[50], b[50]; clrscr();  printf("Enter the first string\n");  scanf(“%s”,a);  printf("Enter the second string\n");  scanf(“%s”,b);  if (strcmp(a,b) == 0) printf("strings are equal.\n"); else  printf("strings are not equal.\n"); getch();  } | #include <stdio.h> #include <conio.h> void main()  {  char a[50], b[50]; int i, temp = 0; clrscr();  printf("Enter the first string:\n");  scanf(“%s”,a);  printf("Enter the second String:\n"); scanf(“%s”,b);  for(i=0; (a[i]!='\0')||(b[i]!='\0'); i++)  {  if(a[i] != b[i])  {  temp = 1; break;  }}  if(temp == 0) printf("strings are equal."); else  printf("strings are not equal."); getch();  } |

|  |  |
| --- | --- |
| **Output:**  Enter the first string  **hai**  Enter the second string  **hai**  strings are equal | |
| **To copy a string** | |
| #include <stdio.h> #include<conio.h> #include <string.h> void main()  {  char a[50], b[50]; clrscr();  printf("Enter a string \n");  scanf(“%s”,a);  strcpy(b, a);  printf("After copy=%s\n",b); getch();  } | #include <stdio.h> #include<conio.h> void main()  {  char a[50], b[50], i; clrscr();  printf("Enter a string"); scanf("%s", a);  for(i=0; a[i]!='\0'; i++)  {  b[i]=a[i];  } b[i]='\0';  printf("After copy =%s\n",b); getch();  } |
| **Output:** Enter a string **Hai**  After copy  **Hai** | |
| **To concatenate two strings** | |
| #include <stdio.h> #include<conio.h> #include <string.h> void main()  {  char a[50], b[50]; clrscr();  printf("Enter the first string\n");  scanf(“%s”,a);  printf("Enter the second string\n");  scanf(“%s”,b); strcat(a,b);  printf("After concatenation:%s\n",a); getch();  } | #include <stdio.h> #include<conio.h> void main()  {  char a[50], b[50], i, j; clrscr();  printf("Enter first string: "); scanf("%s",a);  printf("Enter second string: "); scanf("%s",b);  for(i=0; a[i]!='\0'; ++i);  for(j=0; b[j]!='\0'; ++j, ++i)  {  a[i]=b[j];  } a[i]='\0'; |

|  |  |
| --- | --- |
|  | printf("After concatenation: %s",a); getch();  } |
| **Output:**  Enter the first string  **pani**  Enter the second string  **malar**  After concatenation  **panimalar** | |
| **To reverse the given string** | |
| #include <stdio.h> #include<conio.h> #include <string.h> void main()  {  char a[50]; clrscr();  printf("Enter a string to reverse\n");  scanf(“%s”,a);  strrev(a);  printf("Reversed string is: %s\n",a); getch();  } | #include<stdio.h> #include<conio.h> void main()  {  int l,i ; char a[50]; clrscr();  printf("Enter a String to reverse\n");  scanf(“%s”,a);  for (i = 0; a[i] != '\0'; i++) l++;  printf("Reversed string is:\n"); for(i=l;i>=0;i--)  {  printf("%c",a[i]);  }  getch();  } |
| **Output:**  Enter a string to reverse  **DOG**  Reversed string is:  **GOD** | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **strlwr** |  |  | **Output:**  enter a string HAI  The string in lowercase is hai |
| #include <stdio.h> #include<conio.h> #include <string.h> void main()  {  char a[50]; clrscr();  printf(“enter a string”);  gets(a); scanf(“%s”,a); strlwr(a);  printf("The string in lowercase is: %s\n",a); getch();  } | | | |
| **strupr** | | | | **Output:**  enter a string hai  The string in uppercae is HAI |
| #include <stdio.h> | | | |
| #include<conio.h> | | | |
| #include <string.h> | | | |
| void main() | | | |
| { | | | |
| char a[50]; | | | |
| clrscr(); | | | |
| printf("Enter a string \n"); | | | |
| gets(a); | | | |
| strupr(a); | | | |
| printf("The string in uppercase is: %s\n",a); | | | |
| getch(); | | | |
| } | | | |
| **strset** | | | | **Output:**  Enter a string hai  enter a character a  The string after strset is: aaa |
| #include<stdio.h> | | | |
| #include<conio.h> | | | |
| #include<string.h> | | | |
| void main() | | | |
| { | | | |
| char a[50],b; | | | |
| clrscr(); | | | |
| printf("Enter a string\n"); | | | |
| gets(a); | | | |
| printf("enter a character\n"); | | | |
| scanf("%c",&b); | | | |
| strset(a,b); | | | |
| printf("The string after strset is: %s\n",a); | | | |
| getch(); | | | |
| } | | | |

|  |  |
| --- | --- |
| **strchr/strnchr**  #include<stdio.h> #include<conio.h> #include<string.h> void main()  {  char a[50],b,\*p; clrscr();  printf("Enter a string\n"); gets(a);  printf("enter a character\n"); scanf("%c",&b); p=strchr(a,b);  if(p==NULL)  printf("The character is not found"); else  printf("the character is found"); getch();  } | **Output:**  Enter a string hai  enter a character a  the character is found |
| **strstr**  #include <stdio.h> #include<conio.h> #include <string.h> void main()  {  char a[50],b[10],\*p; clrscr();  printf("Enter a string\n"); gets(a);  printf("enter a substring"); gets(b);  p=strstr(a,b); if(p==NULL)  printf("The substring is not found"); else  printf("the substring is found"); getch();  } | **Output:**  Enter a string welcome  enter a substring come  the substring is found |

##### Programs on strings:

1. To check the given string is palindrome or not
2. Count blank spaces in a paragraph
3. Count number of vowels consonants and digits in a paragraph
4. Count no of occurrences of a particular character in a string
5. Sorting names in alphabetical order

|  |  |
| --- | --- |
| ***To check the given string is palindrome or not***  #include <stdio.h> #include<conio.h> #include <string.h> void main()  {  char a[100], b[100]; clrscr();  printf("Enter the string\n"); scanf("%s",a);  strcpy(b,a); strrev(b);  if (strcmp(a,b) == 0)  printf("Entered string is a palindrome.\n"); else  printf("Entered string is not a palindrome.\n"); getch();  } | ***OUTPUT:***  Enter the string:  **madam**  Entered string is a palindrome. |
| ***Count blank spaces in a paragraph*** #include<stdio.h> #include<conio.h>  void main()  {  char line[10]; int i,count=0; clrscr();  printf("Enter the string\n"); gets(line); for(i=0;line[i]!='\0';i++)  {  if(line[i]==' ') count++;  }  printf("Total no of Spaces : %d",count); getch();  } | ***OUTPUT:***  Enter the string  **God is love**  Total no of Spaces: **2** |
| ***Count number of vowels consonants and digits in a paragraph***  #include<stdio.h> #include<conio.h> void main()  {  char line[150]; int i,v,c,d,s;  v=c=d=s=0; clrscr(); | ***OUTPUT:***  Enter the string **God is love1** Vowels: **4**  Consonants: **5**  Digits: **1**  White spaces:**2** |

|  |  |
| --- | --- |
| printf("Enter a line of string:\n"); gets(line); for(i=0;line[i]!='\0';i++)  {  if(line[i]=='a' || line[i]=='e' || line[i]=='i' || line[i]=='o'  || line[i]=='u' || line[i]=='A' || line[i]=='E' || line[i]=='I'  || line[i]=='O' || line[i]=='U') v++;  else if((line[i]>='a'&& line[i]<='z') || (line[i]>='A'&& line[i]<='Z'))  c++;  else if(line[i]>='0'&&line[i]<='9') d++;  else if (line[i]==' ') s++;  }  printf("Vowels: %d",v); printf("\nConsonants: %d",c); printf("\nDigits: %d",d); printf("\nWhite spaces: %d",s); getch();  } |  |
| ***Count no of occurrences of a particular character in a string***  #include <stdio.h> #include<conio.h> void main()  {  char a[100], ch; int count=0, i; clrscr();  printf("Enter a string\n"); gets(a);  printf("Enter the character to be searched\n"); scanf("%c", &ch);  for(i = 0; a[i]!='\0'; i++)  {  if(a[i]==ch) count++;  }  if(count==0)  printf("Character is not present\n"); else  printf("ocurrences of a character is %d",count); getch();  } | ***OUTPUT:***  Enter a string  **HELLO**  Enter the character to be searched  **L**  No of occurrences is **2** |

|  |  |
| --- | --- |
| ***Program for sorting names:*** #include<stdio.h> #include<conio.h>  void main()  {  int i,j,n;  char a[20][20],temp[20]; clrscr();  printf("Enter the no. of string to be sorted"); scanf("%d",&n);  printf("Enter the strings:"); for(i=0;i<n;i++)  {  scanf("%s",a[i]);  }  for(i=0;i<n;i++)  {  for(j=i+1;j<n;j++)  {  if(strcmp(a[i],a[j])>0)  {  strcpy(temp,a[i]);  strcpy(a[i],a[j]);  strcpy(a[j],temp);  }  }  }  printf("The sorted strings:\n"); for(i=0;i<n;i++)  {  printf("%s\n",a[i]);  }  getch();  } | ***OUTPUT:***  **Enter the no. of string to be sorted:**  4  **Enter the strings:**  viji yazhini hema geetha  **The sorted strings:**  geetha hema viji yazhini |

# Sorting:

Sorting is a process of rearranging the given elements in to either ascending or descending order.

##### Types:

1. Bubble sort
2. Insertion sort
3. selection sort
4. Quick sort
5. Radix sort
6. Heap sort

|  |  |
| --- | --- |
| ***Sorting in ascending and descending order***  #include<stdio.h> #include<conio.h> void main()  {  int a[50], n, i, j, temp; clrscr();  printf("Enter the size of an array"); scanf("%d", &n);  printf("Enter the array elements"); for (i = 0; i < n; i++)  scanf("%d", &a[i]); for (i = 0 ; i < n; i++)  {  for (j = i+1;j<n;j++)  {  if (a[i] > a[j])  {  temp= a[i]; a[i]=a[j]; a[j] = temp;  }  }  }  printf("Array after Sorted in ascending order:\n"); for ( i = 0 ; i < n ; i++ )  printf("%d\n", a[i]);  printf("Array after Sorted in descending order:\n"); for(i=n-1;i>=0;i--)  printf("%d\n", a[i]); getch();  } | ***OUTPUT:***  Enter the size of an array: 5  Enter the array elements: 6  2  0  7  9  Array after Sorted in ascending order:  0  2  6  7  9  Array after Sorted in descending order:  9  7  6  2  0 |

|  |  |
| --- | --- |
| ***Program to find max, min, second largest , second smallest, middle element of an array***  #include<stdio.h> #include<conio.h> void main()  {  int a[50], n, i, j, temp,m; clrscr();  printf("Enter the size of an array"); scanf("%d", &n);  printf("Enter the array elements"); for (i = 0; i < n; i++)  scanf("%d", &a[i]); for (i = 0 ; i < n; i++)  {  for (j = i+1;j<n;j++)  {  if (a[i] > a[j])  {  temp= a[i]; a[i]=a[j]; a[j] = temp;  }  }  }  printf("the sorted array in ascending order\n"); for(i=0;i<n;i++)  printf("%d\n",a[i]);  printf("The greatest element is: %d\n", a[n-1]); printf("The smallest element is: %d\n", a[0]); printf("The second largest element is: %d\n", a[n-2]); printf("The second smallest element is: %d\n", a[1]); m=(0+(n-1)/2);  printf("The middle element is: %d\n", a[m]); getch();  } | ***OUTPUT:***  **Enter the size of an array:**  5  **Enter the array elements:**  6  2  0  7  9  **the sorted array in ascending order**  0  2  6  7  9  **The greatest element is: 9 The smallest element is: 0 The second largest element is :7**  **The second smallest element is :2**  **The middle element is:6** |

***Searching***

Searching is a process of finding a particular element in a given list.

##### Types:

1. Linear search
2. Binary search

**Linear search**

Linear search is also known as sequential search.

##### Advantages:

* The linear search is simple - It is very easy to understand and implement.
* It does not require the data in the array to be stored in any particular order.

##### Disadvantages:

* It has a very poor O(n) general efficiency. That is, the performance of the algorithm scales linearly with the size of the input.
* Linear search thus is considerably slower than many other search methods.

***Linear search:*** #include<stdio.h> #include<conio.h> void main()

{

int a[50], i, n, x, pos=-1; clrscr();

printf("Enter the size of an array\n"); scanf("%d",&n);

printf("Enter the elements of an array"); for (i = 0; i < n; i++)

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

printf("Enter the number to searched\n"); scanf("%d", &x);

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

{

if (a[i] == x)

{

pos=i; break;

}

}

if(pos==-1)

printf("the element is not in the list"); else

printf("The element is found at index %d", pos); getch();

}

##### Output:

Enter the size of an array 4

Enter the elements of an array 2

1

0

7

Enter the number to searched 1

The element is found at index 1

## Binary search:

It uses divide and conquer method.

This method can be applied only if list is in sorted order. List is divided into two halves separated by middle element.

|  |  |  |
| --- | --- | --- |
| Left half | Middle | Right half |

##### Steps:

1. Let x be the element to search.
2. The middle element is tested for the required element .If it is found , then its position is reported else,
3. If x<middle search in the left half, otherwise search in the right half.
4. Repeat step 1, 2 on the selected half , until the element is found, Otherwise report failure(element not found).

**Example:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 4 | 5 | 7 | 11 | 13 | 14 | 17 | 21 |

0 1 2 3 4 5 6 7 8

#### Step1:

F=0, L=8, M=(F+L)/2

Middle position=(0+8)/2=4 Middle element=11

Search element X=14

**Step2:** X<M 14<11 false , search in the right half.

|  |  |  |  |
| --- | --- | --- | --- |
| 13 | 14 | 17 | 21 |

5 6 7 8

F=5, L=8, M= M=(F+L)/2

Middle position=(5+8)/2= 6 Middle element=14 Element found.

|  |  |
| --- | --- |
| **Binary search:** | ***Output:*** |
| #include<stdio.h> | Enter the size of an array |
| #include<conio.h> | 5 |
| void main() | Enter the elements of an array |
| { | 1 |
| int i, first, last, middle, n, x, a[50]; | 5 |
| clrscr(); | 2 |
| printf("Enter the size of an array \n"); | 8 |
| scanf("%d",&n); | 7 |
| printf("Enter the elements of an array"); | Enter the number to search |
| for (i = 0; i < n; i++) | 7 |
| scanf("%d",&a[i]); | 7 found at location 4 |
| printf("Enter the number to searched\n"); |  |
| scanf("%d", &x); |  |
| first=0; |  |
| last=n-1; |  |
| middle = (first+last)/2; |  |
| while (first <= last) |  |
| { |  |
| if (a[middle] == x) |  |
| { |  |
| printf("%d found at location %d.\n", x, middle); |  |
| break; |  |
| } |  |
| else if (a[middle] > x) |  |
| last = middle - 1; |  |
| else |  |
| first = middle + 1; |  |
| middle = (first + last)/2; |  |
| } |  |
| if (first > last) |  |
| printf("Not found! %d is not present in the list.\n", x); |  |
| getch(); |  |
| } |  |