binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials

Catalog



binils.com

binils – Android App

binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials
Binils.com – Free Anna University, Polytechnic, School Study Materials

# ARRAYS AND STRINGS

## INTRODUCTION TO ARRAYS:

An array is a collection of homogeneous (similar) data items that are stored under one common name. Individual data item (array elements) in an array is identified by index or subscript enclosed in square brackets with array name.

- ✓ Array elements can be integers, floating point numbers and so on, but they must be the same type and same storage class.

### Features of Array

- ✓ An array is a derived data type. It is used to represent a collection of elements of the same data type.
- ✓ Array elements are counted from 0 to size-1.
- ✓ The elements can be accessed with array name and the index. The index specifies the position of the element.
- ✓ The elements in an array are stored in continuous memory location. The starting memory location is represented by the array name and it is known as the base address of the Array.

binils.com

### Advantageof Array

1) **CodeOptimization:** Less code to the access the data.
2) **Easy totraversedata**: By using the for loop, we can retrieve the elements of an array easily.
3) **Easy tosortdata**: To sort the elements of array, we need a few lines of code only.
4) **Random Access**: We can access any element randomly using the array.

### Types of array

Arrays can be classified into

- ✓ One Dimensional Array
- ✓ Two Dimensional Array
- ✓ Multi Dimensional Array

binils – Android App

binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials


**ONE DIMENSIONAL ARRAY**

If the array has only one subscript then it is called one dimensional or single dimensional array. An array is a collection of homogeneous (similar) data items that are stored under one common name.

Characteristics of One Dimensional Array

- ✓ Array size must be positive number.
- ✓ Array elements are counted from 0 to size-1.
- ✓ String arrays are terminated with null character ('\0').

**Declaration of an Array**

Arrays must be declared before they are used so that the compiler can allocate space for them in memory.

**Syntax for array declaration**

data type array_name [size];

- The data type specifies the array elements data type.
- Size indicates the maximum number of elements that can be stored in the array.

**Example**

float height [10];

binils.com

The above array declaration represents the array name is height, we can store a maximum of 10 elements and the array elements are floating point data type.

**Different data type declaration of array**

1. int a[5];

| | | | | |
|---|---|---|---|---|
| a[0] | a[1] | a[2] | a[3] | a[4] |

2. float b[5];

| | | | | |
|---|---|---|---|---|
| b[0] | b[1] | b[2] | b[3] | b[4] |

3. char m[5];

| | | | | |
|---|---|---|---|---|
| m[0] | m[1] | m[2] | m[3] | m[4] |

***Different data type declaration of array***


binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials


**Array Initialization**

The array elements can be initialized when they are declared otherwise they will take garbage values.

**Compile time initialization**

Arrays can be initialized at compile time.

**Syntax:**

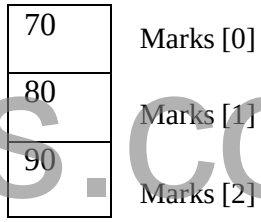
data type array_name [size] = {value 0, value 1, . . . , value n-1)

The initialized values are specified within curly braces separated by commas.

**Example:**

int Marks [3] = {70, 80, 90};

This statement declares the variable Marks as an array of 3 elements and will be assigned the values specified in list as below.



binils.com

**Array Initialization**

Like ordinary variables, the values to the array can be initialized as follows.

```
int Marks[3];
Marks [0] = 70;
Marks [1] = 80;
Marks [2] = 90;
```

Character array can be initialized as follows:

```
char gender[2] = {'M','F'};
```

**Runtime initialization**

Arrays can be initialized at run time.

**Example:**

```
int a[2];
scanf("%d%d",&a[0],&a[1]);
```


binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials


**Program: Calculate the average marks of the student**

```
# include<stdio.h>
#include<conio.h>
void main()
{
        int m[5],i,sum=0,n;
        float avg;
        printf("enter number of subject \n");
        scanf("%d",&n);
        printf("enter marks \n");
        for(i=0;i<n;i++)
        {
                scanf("%d",&m[i]);
        }
        for(i=0;i<n;i++)
        sum=sum+m[i];
        avg=(float)sum/n;
        printf("average=%f",avg);
        getch()
}
```

binils.com

**Output**:

```
Enter number of subject
5
Enter marks of students
55
60
78
85
90
Average=73.6
```


binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials
Binils.com – Free Anna University, Polytechnic, School Study Materials

## TWO DIMENSIONAL ARRAYS

If the array has two subscripts then it is called two dimensional array or matrix. Two dimensional arrays are used in situation where a table of values needs to be stored. A 2D array is an array of 1-D arrays and can be visualized as a plane that has rows and columns.

**Declaration**

Two dimensional arrays must be declared before they are used so that the compiler can allocate space for them in memory.

**Syntax for declaration**

datatype array_name [row size] [col size];

- The data type specifies the array elements data type.
- rowsize indicates the size of row
- colsize indicates the size of column

**Example**

int matrix[5] [5];

char name[10] [20]; // 10 rows 20 columns

binils.com

The above array declaration represents the array name is height, we can store a maximum of 10 elements and the array elements are floating point data type.

**a**

**a[3][5]**

| | | | | |
|---|---|---|---|---|
| [0][0] | [0][1] | [0][2] | [0][3] | [0][4] |
| [1][0] | [1][1] | [1][2] | [1][3] | [1][4] |
| [2][0] | [2][1] | [2][2] | [2][3] | [2][4] |

***Memory layout representation***

**Initialization**

The array elements can be initialized when they are declared otherwise they will take garbage values.

**Compile time initialization**

Arrays can be initialized at compile time.

**Syntax**

datatype array_name [row size] [col size];= {value 0, value 1, . . . , value n-1};

binils – Android App



binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials


The initialized values are specified within curly braces separated by commas.

**Example:**

```
int matrix[3][5] = { {2, 6, 7,8,9} , {10, -50, 3,5,6},{2,4,6,8,20} };
```

| | | | | |
|---|---|---|---|---|
| 2 | 6 | 7 | 8 | 9 |
| 10 | -50 | 3 | 5 | 6 |
| 2 | 4 | 6 | 8 | 20 |

***Fig. 2.4: Memory Representation of 2D Array***

**Runtime initialization**

Arrays can be initialized at run time.

**Example:**

```
int a[2][2];
scanf("%d%d",&a[0][0],&a[0][1],&a[1][0],&a[1][1]);
```

binils.com

**Program: Find the addition of two matrix**

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int a[25][25],b[25][25], c[25][25], i, j m, n;
    clrscr();
    printf("\n Enter the rows and columns of two matrices… ");
    scanf("%d %d ", &m, &n)
    printf("\n Enter the elements of A matrix…");
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            scanf("%d",&a[i][j]);
    printf("\n Enter the elements of B matrix…");
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
```


binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials


```
                scanf("%d", &b[i][j]);
        for(i=0;i<m;i++)
                for(j=0;j<n;j++)
                        c[i][j]=a[i][j] + b[i][j];
        printf("\n The addition of two matrices");
        for(i=0;i<m;i++)
        {
                printf("\n");
                for(j=0;j<n;j++)
                {
                        printf("\t %d",c[i][j]);
                }
        }
        getch();
}
```

binils.com

**Output:**

```
Enter the rows and columns of two matrices… 3 3
Enter the elements of A matrix
 1      2      3
4       5      6
7       8      9
Enter the elements of B matrix
1       2      3
4       5      6
7       8      9
The addition of two matrixes
2       4      6
8       10     12
14      16     18
```


binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials


**STRING Declaration & Initialization**

String is a collection of characters. In C language array of characters are called string. It is enclosed within double quotes. E.g. “INDIA” is a string. Each character of string occupies 1 byte of memory. The last character is always „\0“.

**Declaration of a String**

Strings can be declared like a one dimensional array.

**Syntax:**

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

**Example:**

```
char name[30];
char dept[20];
```

binils.com

**String Initialization**

The string can be initialized as follows:

```
char dept[10] = “CSE”;
```

OR

```
char dept[] = {„C“, „S“, „E“,“\0“};
```

In the above example, „\0“ is a null character and specifies end of the string. Here string is assigned character by character.


binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials


**Example Program :**

```
#include <stdio.h>
int main() {
char name[10];
int age;
printf("Enter your first name and age: \n");
scanf("%s %d", name, &age);
printf("You entered: %s %d",name,age);
}
```

**Output:**

```
Enter your first name and age:
Hamsini 21
```

binils.com


binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials


**STRING OPERATIONS - Length**

**String Handling / Manipulation Function:**

For string handling, we must include <string.h> header file in program

| Function | Purpose |
|---|---|
| strlen () | Used to find length of a string |
| strcpy () | Used to copy one string to another |
| strcat () | Used to concatenate two strings |
| strcmp () | Used to compare characters of two strings |
| strlwr () | Convert strings into lower case |
| strupr () | Convert strings into upper case |
| strrev () | Used to reverse a string |

binils.com

**String Handling Function**


binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials


**String Length**

strlen() function is used to find the length of a string. It is used to count and return the number of characters present in a string. The terminating character („\0") is not counted.

**Syntax**

temp_variable = strlen(string_name);

**Example**

s= "hai";

strlen(s)-> returns 3.

**Program:**

```
# include<stdio.h>
#include<conio.h>
# include <string.h>
void main()
{
        char name[10];
        printf("Enter string:");
        scanf("%s",name);
        int length;
        length = strlen(name);
        printf("\n String length of %s is %d",  name, length);
        getch();
}
```

binils.com

**Output:**

Enter string: APPLE

String length of APPLE is 5


binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials
Binils.com – Free Anna University, Polytechnic, School Study Materials

**STRING OPERATIONS –Copy, Compare, Concatenation**

**String Copy**

It copies the source string to the destination string or used to copy the contents of one string to another string variable

**Syntax**

strcpy(destination,source);

**Example**

s1="hai";

s2= "welcome";

strcpy(s1,s2); // s2 is copied to s1. i.e. s1=welcome.

**Program:**

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char source[10];
printf("Enter string:");
scanf("%s",source);
char target[10];
strcpy(target, source);
printf("\n Source string is %s", source);
printf("Target string is %s, target);
getch();
}
```

binils.com

**Output:**

Enter string : COMPUTER

Source string is COMPUTER

Target string is COMPUTER


binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials


**String concatenation**

strcat() is used to concatenate or combine two strings together.

**Syntax**:

strcat (string1, string2);

String2 is concatenated at the end of string1 and the result is stored in string1.

**Example**

s1="hai ";

s2= "welcome";

strcat(s1,s2); // s2 is joined with s1. Now s1 is hai welcome.

**Program:**

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char source[10];
    printf("Enter string1:");
    scanf("%s",source);
    char target[10];
    printf("Enter string 2:");
    scanf("%s",target);
    strcat(source, target);
    printf("After concatenation source string is : %s", source);
    getch();
}
```

binils.com

**Output:**


binils - Anna University App on Play Store

Enter string1 : Computer

Enter string2 : Programming

After concatenation source string is: Computer Programming

**String Compare**

strcmp() function compares two strings to check whether they are same or different. The two strings are compared character by character until end of one string is reached or a mismatch character found.

- If two strings are identical, strcmp() returns a value zero
- If they are not equal it returns the numeric difference between the first non-matching characters.
- if the strcmp() returns positive then string1 is greater and negative means string2 is greater.

**Syntax:**

strcmp(string1, string2);

binils.com

**Program:**

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char source[10];
    printf("Enter string1:");
    scanf("%s",source);
    char target[10];
    printf("Enter string 2:");
    scanf("%s",target);
    int diff;
    diff = strcmp (source, target);
```

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials


```
        if (diff == 0 )
                printf("Both strings are identical");
        else
                printf("Both strings are not identical");
        getch();
}
```

**Output:**

Enter String 1: Computer

Enter String 2: Programming

Both strings are not identical

binils.com


binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials

**SORTING**

Sorting is the process of arranging elements either in ascending or in descending order. Some of the sorting Methods are,

- Selection Sort
- Bubble Sort
- Merge sort
- Quick sort

**Selection Sort**

It finds the smallest element in the list & swaps it with the element present at the head of the list. It is a very simple and natural way of sorting a list.

**Steps:**

- In selection sort the first element is compared with the remaining elements.
- If the first element is larger than the other elements, it should be interchanged.
- In the second iteration, the second element is compared with the following elements and interchange if not in order.
- This step is continued for various iterations, until the elements are sorted in an order.

binils.com

**Example :** (44, 33, 55, 22, 11)

**Iteration 1:** First compare first element with all other element.

44, 33, 55, 22, 11 Compare 44 & 33. Not in order. So swap.

33, 44, 55, 22, 11 Compare 33 & 55. It is in order. So don't swap.

33, 44, 55, 22, 11 Compare 33 & 22. Not in order. So swap.

22, 44, 55, 33, 11 Compare 22 & 11. Not in order. So swap.

11, 44, 55, 33, 22 Now first element is in correct order.

**Iteration 2:** Now compare second element with all other element.

11, 44, 55, 33, 22 Compare 44 & 55. It is in order . So don't swap.

11, 44, 55, 33, 22 Compare 44 & 33. Not in order . So swap.

11, 33, 55, 44, 22 Compare 33 & 22. Not in order . So swap.

11, 22, 55, 44, 33 Now first two elements are in correct order.

**Iteration 3:** Now compare third element with all other element.

11, 22, 55, 44, 33 Compare 55 & 44. Not in order . So swap.

11, 22, 44, 55, 33 Compare 44 & 33. Not in order . So swap.

binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials

11, 22, 33, 55, 44 Now first three elements are in correct order.

**Iteration 4:** Now compare fourth element with all other element.

11, 22, 33, 55, 44 Compare 55 & 44. Not in order . So swap.

11, 22, 33. 44, 55 Now first 4 elements are in correct order.

Balance only one element is there. So sorting is over.

**Program:**

```
#include <stdio.h>
#include<conio.h>
void main()
{
    int i, j, temp, n, a[10];
    printf("Enter the value of N \n");
    scanf("%d", &n);
    printf("Enter the numbers \n");
    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 numbers arranged in ascending order are given below \n");
  for (i = 0; i < n; i++)
    printf("%d\n", a[i]);
```

binils.com

binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials

```
  printf("The numbers arranged in descending order are given below \n");
  for(i=n-1;i>=0;i--)
        printf("%d\n",a[i]);
   getch();
}
```

binils.com

binils - Anna University App on Play Store

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials


**SEARCHING**

Searching is to find a particular element in a list of elements. Following are some of the searching methods.

1. Linear Search
2. Binary Search

**Linear Search (or) Sequential Search:**

This is the simplest method of searching a data in an array. This can be applied in an unsorted array. Its time complexity is O ( n ).

**Steps:**

- In Linear search, we start the search from first location.
- If data found we stop otherwise search continues with next location.
- The above step is repeated until we reach the last location.
- Whenever we reach end of the list, the data is not found.

binils.com

**Example 1:**

Consider the array elements: 3, 15, 12, 13, 8 and

Element to be searched = 13


binils - Anna University App on Play Store

**Steps:**

First we compare 13 with first element 3 - No match

Then compare 13 with second element 15 – No match

Then compare 13 with third element 12 – No match

Then compare 13 with fourth element 13 – Match.

Thus, the data found at location 4.

**Example 2:**

Consider the array elements: 3, 15, 12, 13, 8 and

Element to be searched (key) = 7

**Steps:**

First we compare 7 with first element 3 - No match

Then compare 7 with second element 15 – No match

Then compare 7 with third element 12 – No match

Then compare 7 with fourth element 13 – No match.

Then compare 7 with fifth element 8 – No match.

Now we reached the end of the list. So we say that "data not found".

binils.com

**Program: Linear Search**

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int a[10],i,n,m,c=0;
   printf("Enter the size of an array: ");
   scanf("%d",&n);
   printf("Enter the elements of the array: ");
   for(i=0;i<=n-1;i++)
        scanf("%d",&a[i]);
   printf("Enter the number to be searched: ");
        scanf("%d",&m);
   for(i=0;i<=n-1;i++)
   {
```

```
        if(a[i]==m)
        {
            printf("Element is in the position %d\n",i+1);
            c=1;
            break;
        }
    }
    if(c==0)
        printf("The number is not in the list");
    getch();
}
```

**Output:**

```
Enter the size of an array: 4
Enter the elements of the array: 4 3 5 1
Enter the number to be search: 5
Element is in the position 3
```

binils.com

binils.com - Anna University, Polytechnic & Schools
Free PDF Study Materials


**Binary Search:**

Binary search can be applied only on sorted data. It is faster than linear search. Its time complexity is O(log n)

**Steps:**

- First check the search element at the middle of the list
- If the middle element is the search key, then the data found. Stop the searching.
- If the key to be searched is smaller than the middle element then continue with the bottom half of the list.
- If the key to be searched is greater than the middle element then continue with the top half of the list
- Repeat the above steps till the sub array not possible further divide.

**Example:**

Consider the array elements

10, 15, 17, 23, 45, 60, 75

binils.com


binils - Anna University App on Play Store

And the search element key = 15

**1st Iteration:**

High = 7, Low =1

Mid = (low + high) / 2 = (1 + 7) / 2 = 8/2 = 4

Array [mid] = Array[4] = 23

Here 15 < 23 search continues on the left side of the array

Therefore new High = mid -1 = 4 – 1 = 3

The sub list = 10, 15, 17

**2nd iteration:**

High = 3, Low = 1

Mid = (low + high) / 2 = (1 + 3) / 2 = 4 / 2 = 2

Array[mid] = Array[2] = 15

Here Array[mid] = key. i.e., 15 = 15

Thus data found at mid. i.e., at location 2.

**Program : Binary Search**

binils.com

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int A[10], i, n, key,c=0;
    int mid, low, high;
    printf("Enter the size of an array:");
    scanf("%d", &n);
    printf("Enter the elements in ascending order :");
    for( i = 1; i <= n; i++)
        scanf("%d", &A[i]);
    low = 1; high = n;
    printf("Enter the number to be searched:");
        scanf("%d", &key);
    while ( low <= high)
    {
```

```
            mid = (low + high) / 2;
            if ( A[mid] == key)
                  c=1;
                  break;
            else if ( A[mid] < Key )
                  low = mid+ 1;
            else
                  high = mid – 1;
      }

      if(c==0)
            printf("The number is not found.");
      else
            printf("The number is found.");
      getch();
}
```

binils.com

**Output:**

Enter the size of an array: 5

Enter the elements in ascending order: 4 7 8 11 21

Enter the number to be searched: 11

The number is found.