



PROGRAMMING

FOR PROBLEM SOLVING

{ C LANGUAGE }



Module :- 4

Start

Topic :-

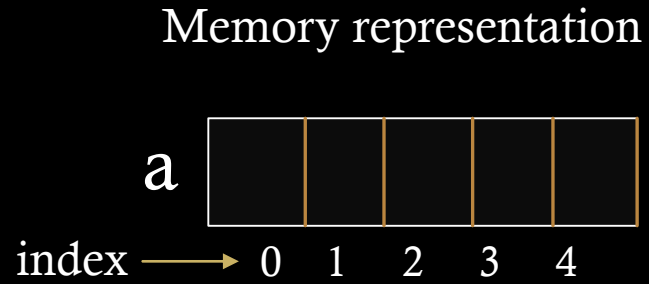
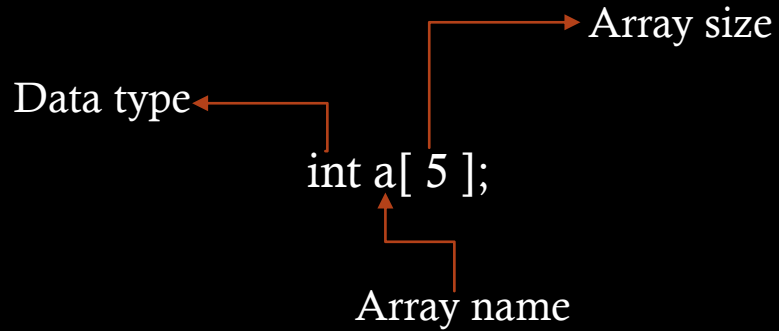
- ◇ Array declaration and initialization
- ◇ Bound checking arrays (1-D , 2-D)
- ◇ Character arrays
- ◇ strings

Single Dimension Array

◇ **Array :-**

- It is a collection of data of same data type
- It is used to store group of data simultaneously
- It can store data of same data type means an integer array can store only integer value , character array can store only character value and so on
- We can not fetch data from array directly therefore we use index point
- The indexing of array always start with 0
- Index value is always an integer number
- Array may be of any data type like int , char , float etc.

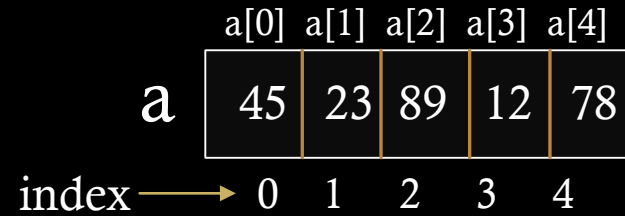
Syntax



- Here `a` is the name of array
- `int` is the data type of array
- Size of array is 5 means we can store maximum 5 values in this array

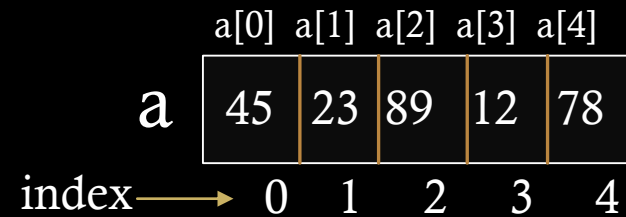
Initialization of array method 1

```
int a[5] = { 45,23,89,12,78 };
```



Initialization of array method 2

```
int a[5];  
a[0]=45;  
a[1]=23;  
a[2]=89;  
a[3]=12;  
a[4]=78;
```



Printing of array element method 1

```
//Printing of array element method 1
#include<stdio.h>
int main()
{
    int a[5]={ 45,23,89,12,78 };
    printf("value at a[0]=%d\n",a[0]);
    printf("value at a[1]=%d\n",a[1]);
    printf("value at a[2]=%d\n",a[2]);
    printf("value at a[3]=%d\n",a[3]);
    printf("value at a[4]=%d\n",a[4]);
}

// ### Output ###
// value at a[0]=45
// value at a[1]=23
// value at a[2]=89
// value at a[3]=12
// value at a[4]=78
```

Printing of array element using loop

```
//Printing of array element using loop
#include<stdio.h>
int main()
{
    int a[5]={ 45,23,89,12,78 };
    for(int i=0;i<=4;i++)
    {
        printf("value at a[%d]=%d\n",i,a[i]);
    }
}
// ### Output ###
// value at a[0]=45
// value at a[1]=23
// value at a[2]=89
// value at a[3]=12
// value at a[4]=78
```

User input in array

```
//User input in array
#include<stdio.h>
int main()
{
    int a[3];//declaring array
    printf("Enter element 1=");
    scanf("%d",&a[0]);//taking input at index 0
    printf("Enter element 2=");
    scanf("%d",&a[1]);//taking input at index 1
    printf("Enter element 3=");
    scanf("%d",&a[2]);//taking input at index 2
    for(int i=0;i<=2;i++)
    {
        printf("value at a[%d]=%d\n",i,a[i]);
    }
}
// ### Output ###
// Enter element 1=12
// Enter element 2=45
// Enter element 3=17
// value at a[0]=12
// value at a[1]=45
// value at a[2]=17
```





User input in array using loop

```
//User input in array using loop
#include<stdio.h>
int main()
{
    int a[3]; //declaring array
    for(int j=0;j<=2;j++)
    {
        printf("Enter element %d=", (j+1));
        scanf("%d",&a[j]); //taking user input
    }
    for(int i=0;i<=2;i++)
    {
        printf("value at a[%d]=%d\n", i, a[i]);
    }
}

// ### Output ###
// Enter element 1=12
// Enter element 2=45
// Enter element 3=17
// value at a[0]=12
// value at a[1]=45
// value at a[2]=17
```

Examples

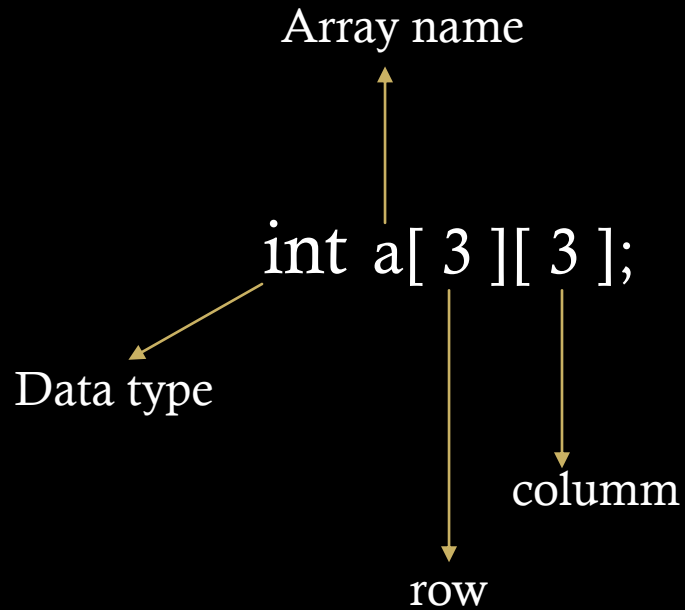
1. Print all the element of array
2. User input in single dimension array
3. Print all the element of array in reverse order
4. Find sum of all the element of array
5. Addition of single dimension array
6. Print all the even number of single dimension array
7. Print all the odd number of single dimension array
8. Find maximum value from single dimension array
9. Find minimum value from single dimension array
10. Searching in single dimension array
11. Copy the element of one array into another
12. Merge the element of first and second into third array
13. Linear search in array
14. Binary search in array

Double dimension array in c



- ◇ It is a collection of data of same data type
- ◇ It is used to store group of data simultaneously
- ◇ It can store data of same data type means an integer array can store only character value and so on
- ◇ We can not fetch data from array directly therefore we use index point
- ◇ The indexing of array always start with 0
- ◇ Index value is always an integer number
- ◇ Array may be of any data type like int , char , float etc..

Syntax



	Col 1	Col 2	Col 3
Row 1	<code>a[0][0]</code>	<code>a[0][1]</code>	<code>a[0][2]</code>
Row 2	<code>a[1][0]</code>	<code>a[1][1]</code>	<code>a[1][2]</code>
Row 3	<code>a[2][0]</code>	<code>a[2][1]</code>	<code>a[2][2]</code>

- ◆ Here a is the name of array
- ◆ Int is the data type of array
- ◆ Size of array is 3x3 means we can store maximum 9 values in this array

Initialization of array method 1

Int a[3][3] = { {40,50,60},{10,20,30},{70,80,90} }

	Col 1	Col 2	Col 3
Row 1	40	50	60
Row 2	10	20	30
Row 3	70	80	90

	Col 1	Col 2	Col 3
Row 1	a[0][0]	a[0][1]	a[0][2]
Row 2	a[1][0]	a[1][1]	a[1][2]
Row 3	a[2][0]	a[2][1]	a[2][2]

Initialization of array method 2



```
Int a[3][3];
```

```
a[0][0]=40;
```

```
a[0][1]=50;
```

```
a[0][2]=60;
```

```
a[1][0]=10;
```

```
a[1][1]=20;
```

```
a[1][2]=40;
```

```
a[2][0]=70;
```

```
a[2][1]=80;
```

```
a[2][2]=90;
```

Col 1 Col 2 Col 3

↓ ↓ ↓

Row 1 →	40	50	60
Row 2 →	10	20	30
Row 3 →	70	80	90

Col 1 Col 2 Col 3

↓ ↓ ↓

Row 1 →	a[0][0]	a[0][1]	a[0][2]
Row 2 →	a[1][0]	a[1][1]	a[1][2]
Row 3 →	a[2][0]	a[2][1]	a[2][2]

Printing of array element method 1



```
//Printing of array element method 1
#include<stdio.h>
int main()
{
    int a[3][3]={{40,50,60},{10,20,30},{70,80,90}}; //Initializing array
    printf("value at a[0][0]=%d\n",a[0][0]);
    printf("value at a[0][1]=%d\n",a[0][1]);
    printf("value at a[0][2]=%d\n",a[0][2]);
    printf("value at a[1][0]=%d\n",a[1][0]);
    printf("value at a[1][1]=%d\n",a[1][1]);
    printf("value at a[1][2]=%d\n",a[1][2]);
    printf("value at a[2][0]=%d\n",a[2][0]);
    printf("value at a[2][1]=%d\n",a[2][1]);
    printf("value at a[2][2]=%d\n",a[2][2]);
}

// ### Output ###
// value at a[0][0]=40
// value at a[0][1]=50
// value at a[0][2]=60
// value at a[1][0]=10
// value at a[1][1]=20
// value at a[1][2]=30
// value at a[2][0]=70
// value at a[2][1]=80
// value at a[2][2]=90
```

Printing of array element method 2 loop



```
//Printing of array element method 2 loop
#include<stdio.h>
int main()
{
    int a[3][3]={{40,50,60},{10,20,30},{70,80,90}}; //Initializing array
    for(int i=0;i<=2;i++)
    {
        for(int j=0;j<=2;j++)
        {
            printf("%d ",a[i][j]);
        }
        printf("\n");
    }
}
// ### Output ###
// 40  50  60
// 10  20  30
// 70  80  90
```

User input in array using loop



```
//User input in array using loop
#include<stdio.h>
int main()
{
    int a[3][3]; //Declaring array
    int i,j;
    for(i=0;i<=2;i++)
    {
        for(j=0;j<=2;j++)
        {
            printf("Enter element at a[%d][%d]=",i,j);
            scanf("%d",&a[i][j]);
        }
    }
    for(i=0;i<=2;i++)
    {
        for(j=0;j<=2;j++)
        {
            printf("%d ",a[i][j]);
        }
        printf("\n");
    }
}

// ### Output ###
// Enter value at a[0][0]=4
// Enter value at a[0][1]=5
// Enter value at a[0][2]=6
// Enter value at a[1][0]=1
// Enter value at a[1][1]=2
// Enter value at a[1][2]=3
// Enter value at a[2][0]=7
// Enter value at a[2][1]=8
// Enter value at a[2][2]=9
// 4 5 6
// 1 2 3
// 7 8 9
```

Example

1. Print all the element of double dimension array
2. User input in single double dimension array
3. Find sum of all the element of array
4. Addition of double dimension array
5. Subtraction of double dimension array
6. Multiplication of double dimension array
7. Transpose of double dimension array



String Function In C Language

- ◇ String is a collection of character
- ◇ String is a sequence of characters terminated with a null character '\0'
- ◇ C does not support string data type therefore char data type is used to make string
- ◇ String in c is stored in single dimension character array
- ◇ There are many predefined string function character array
- ◇ All the string function are predefined in string.h header file

String Declaration Syntax :- **char** *string_name* *size*

String Declaration and Initialization Syntax :-

```
char str[ ] = "KKScomputerScience";  
char str[50] = "KKScomputerScience";  
char str[ ] = { 'K','K','S', '\0'};  
char str[14] = {'K','K','S', '\0'};
```



Function name

Description

<code>strlen(s);</code>	It is used to get the length of string.
<code>strcpy(s1,s2);</code>	It is used to copy the string here string s2 is copied into string s1.
<code>strcat(s1,s2);</code> s1.	It is used to add two string here string s2 concatenates at the end of string s1.
<code>strcmp(s1,s2);</code>	It is used to compare two string. This function returns 0 if string s1 and string s2 are same. This function returns negative value if s1 < s2. This function returns positive value if s1 > s2 .
<code>strrev(s);</code>	It is used to reverse the string.
<code>strupr(s);</code>	It is used to convert all the character into upper alphabet.
<code>strlwr(s);</code>	It is used to convert all the character into lower alphabet.
<code>strcmpi(s1,s2);</code>	It is also used to compare two string but it ignores the case sensitivity means 'a' and 'A' are treated as same.

strlen(s)



```
//strlen(s)
#include<stdio.h>
#include<string.h>
int main()
{
    char name[200]="Easy";
    printf("%d",strlen(name));
    return 0;
}

// ### Output ###
// 4
//because there is 4 character in Easy
```

strcpy(s1,s2)



```
//strcpy(s1,s2)
#include<stdio.h>
#include<string.h>
int main()
{
    char name1[200]="Radha";
    char name2[200]="Rani";
    strcpy(name1,name2);
    printf("%s",name1);
    return 0;
}

// ### Output ###
// Rani
```

strcmp(s1,s2)



```
//strcmp(s1,s2)
#include<stdio.h>
#include<string.h>
int main()
{
    char s1[200]="Easy";
    char s2[200]="Easy";
    if(strcmp(s1,s2)==0)
    {
        printf("string s1 and string s2 are same.");
    }
    else
    {
        printf("string s1 and string s2 are not same.");
    }
    return 0;
}

// ### Output ###
// string s1 and string s2 are same
```

strcat(s1,s2)



```
//strcat(s1,s2)
#include<stdio.h>
#include<string.h>
int main()
{
    char s1[200]="Easy";
    char s2[200]=" Softwares";
    printf("%s",strcat(s1,s2));
    return 0;
}

// ### Output ###
// Easy Softwares
```

strupr(s)



```
//strupr(s)
#include<stdio.h>
#include<string.h>
int main()
{
    char s[200]="Easy";
    printf("%s",strupr(s));
    return 0;
}

// ### Output ###
// EASY
```

strlwr(s)



```
//strlwr(s)
#include<stdio.h>
#include<string.h>
int main()
{
    char s[200]="Easy";
    printf("%s",strlwr(s));
    return 0;
}

// ### Output ###
// easy
```

strrev(s)



```
//strrev(s)
#include<stdio.h>
#include<string.h>
int main()
{
    char s[200]="ABCD";
    printf("%s",strrev(s));
    return 0;
}

// ### Output ###
// DCBA
```

Example



1. Check mobile number length is correct or not
2. Logging program using string
3. Check given string is palindrome or not
4. Reverse string without using function
5. Count length of string without using function
6. Count only alphabets in a string
7. Count only digits in a string
8. Count only special symbol in a string
9. Count total space in a string
10. Count number of vowel and consonant in a string



Thank You !!

Dhanybad !!

Shukriya !!