MARUDHARKESARIJAINCOLLEGEforWOMEN

PGDEPARTMENTOFCOMPUTERAPPLICATIONS

CLASS:I-BCA

SUBJECT NAME:STRUCTUREDPROGRAMMING LANGUAGE IN C SUBJECT

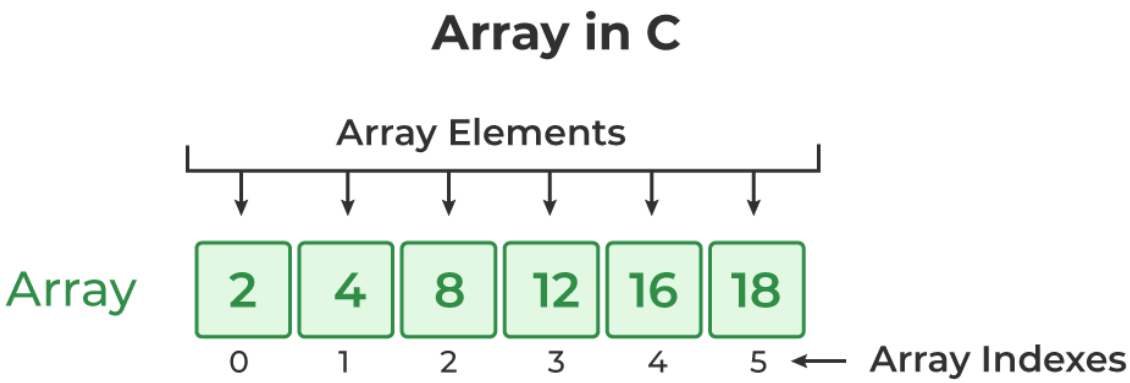
CODE: 23UFC14

UNITIII

Arrays:Declarationandaccessing ofone&two-dimensionalarrays,initializing twoDimensionalarrays, multidimensional arrays.

WhatisArrayinC?

An array in C is a fixed-size collection of similar data items stored in contiguous memory locations. It can be used to store the collectionof primitive data types such as int, char, float, etc., and also derived and user-defined data types such as pointers, structures, etc.



CArrayDeclaration

In C, we havetodeclarethe arraylikeanyothervariablebeforeusingit.We can declare an array by specifying its name, the type of its elements, and the size of its dimensions. When we declare anarray in C, the compiler allocates the memory block of the specified size to the array name.

SyntaxofArrayDeclaration

data_typearray_name[size]; or
data_typearray_name[*size1*][*size2*]...[*sizeN*];

whereNisthenumberofdimensions.

TypesofArrayin C

There aretwotypesofarraysbasedonthenumberofdimensionsithas.Theyare asfollows:

1. One DimensionalArrays(1DArray)
2. MultidimensionalArrays

AdvantageofCArray

1) CodeOptimization:Lesscodetotheaccessthedata.

2) Easeoftraversing:Byusingtheforloop,wecanretrievetheelementsofanarrayeasily.

3) Easeofsorting:Tosorttheelementsofthearray,weneeda fewlinesofcodeonly.

4) RandomAccess:Wecanaccessanyelementrandomlyusingthearray.

DisadvantageofCArray

1) Fixed Size: Whatever size, we define at the time of declaration ofthe array, we can't exceed thelimit. So, it doesn't grow the size dynamically likeLinkedList which we willlearnlater.

DECLARATIONANDACCESSINGOFONE&TWO-DIMENSIONALARRAYS

Todeclarea1DarrayinC,you needto specifythedata type,followedbythearraynameandthe size of thearrayin square brackets. For example, to declareaninteger array named 'numbers' with a size of5, you would write: int numbers[5];

Syntax:data-typearr_name[array_size];

RulesforDeclaringOneDimensionalArray

1. Anarrayvariablemustbedeclaredbeforebeingusedina program.
2. The declaration must have a data type(int, float, char, double, etc.), variable name, andsubscript.
3. The subscript represents the size of the array. If the size is declared as 10, programmers can store 10 elements.
4. Anarrayindexalwaysstartsfrom0.For example, ifanarrayvariableisdeclareds[10], it ranges from 0 to 9.
5. Eacharray elementisstoredinaseparatememorylocation.

INITIALIZATIONOFONE-DIMENSIONALARRAYINC

Anarraycanbeinitializedatthefollowingstates:

1. Atcompilingtime(staticinitialization)
2. DynamicInitialization

COMPILINGTIMEINITIALIZATION:

Thecompile-timeinitializationmeansthearrayoftheelementsisinitializedwhentheprogramis written or array declaration.

Syntax:data_typearray_name[array_size]=(listofelementsofanarray); Example:
int n[5]={0, 1, 2, 3, 4};

ONEDIMENSIONALARRAYINC

Arraysarea fundamental concept inprogramming,andtheycomeindifferent dimensions. ***One-dimensionalarrays***,also known as single arrays, arearrays with only one dimension or a single row. In this article, we'll dive deep into one-dimensional arrays in C programming language, including their ***syntax, examples***, and ***output***.

SyntaxofOne-DimensionalArrayinC

Thesyntaxofa***one-dimensionalarray***inCprogramminglanguageisasfollows: dataType

arrayName[arraySize];

- **dataType** specifiesthedatatypeofthearray.Itcanbeanyvaliddatatypein C programming language, such as ***int, float, char, double***, etc.
- **arrayName**isthenameofthearray,whichisusedtorefer tothearrayintheprogram.
- **arraySize** specifiesthenumberofelements inthearray.Itmustbea ***positiveinteger value***.

ExampleofOne-DimensionalArrayinC

Let's take a simple example of a one-dimensional array in C programming language to understand its syntax and usage.

```
#include<stdio.h>
 intmain(){
   intnumbers[5]={10,20,30,40,50};
for(inti=0;i<5;i++) {
printf("numbers[%d]=%d\n",i,numbers[i]);
  }
    return0;
}
```

OUTPUT:

```
numbers[0]=10
numbers[1]=20
numbers[2]=30
numbers[3]=40
numbers[4]=50
```

Explanation:

Intheaboveexample,wehavedeclaredaone-dimensionalarrayofintegersnamed***numbers***. Thearraycontains fiveelements,andeachelement isinitializedwitha value.

Wehave used a***for loop***toiterate overthe elements of the array and printtheir values using the ***printf*** function.

INITIALIZINGONEDIMENSIONALARRAYINC

InCprogramminglanguage,we caninitializea one-dimensionalarraywhile declaringit or later in the program. We can initialize a one-dimensional array while declaring it by using the following syntax:

```
dataTypearrayName[arraySize]={element1,element2,...,elementN};
```

"dataType'specifiesthedatatypeofthearray.

- **"arrayName'**isthenameofthearray.

- **"arraySize'**specifiesthenumberofelementsinthearray.

- **"{element1, element2, ..., elementN}'**specifies the values of the elements in the array. The number of elements must be equal to ***"arraySize'***.

ExampleofInitializingOneDimensionalArrayinC

Let'stakeanexampletounderstandhowtoinitializea one-dimensionalarrayinCprogramming language.

```
#include
<stdio.h>int main()
{
  intnumbers[5]={10,20,30,40,50}; for(int
    i=0; i<5; i++) {
     printf("numbers[%d]=%d\n",i,numbers[i]);
  }
   return0;
```

}

Outputofthe code:

numbers[0]=10

numbers[1]=20

numbers[2]=30

numbers[3]=40

numbers[4]=50

Explanation:

Intheaboveexample, wehavedeclaredaone-dimensionalarrayofintegersnamednumbersand initialized it withthe values {10,20,30,40,50}.We haveused a for loop to iterate over the elements of the array and print their valuesusing the printf function.

Wecanalsoinitializea one-dimensionalarraylaterintheprogrambyassigningvaluestoits elements using the following syntax:

arrayName[index]=value;

- arrayNameisthenameofthearray.

- indexistheindexoftheelementwewanttoassigna valueto.

- valueisthevaluewewanttoassigntotheelement.

INITIALIZINGTWO-DIMENSIONALARRAYS

The two-dimensional array can be defined as an array of arrays. The 2D array is organized as matrices which can be represented as the collection of rows and columns. However, 2D arrays are created to implement a relational database lookalike data structure. It provides ease of holding the bulk of data at once which can be passed to any number of functions wherever required.

DeclarationoftwodimensionalArrayinC

Thesyntaxtodeclarethe2Darrayisgivenbelow. data_type

array_name[rows][columns];

INITIALIZATIONOF2DARRAYINC

Inthe1D array, we don't needto specifythe size of the array ifthe declarationand initialization are being done simultaneously. However, this will not work with 2D arrays. We will have to defineatleastthe seconddimension ofthearray.Thetwo-dimensionalarraycanbedeclaredand defined in the following way.

intarr[4][3]={{1,2,3},{2,3,4},{3,4,5},{4,5,6}};

Two-dimensionalarrayexampleinC

```
#include<stdio.h>

intmain(){

inti=0,j=0;

intarr[4][3]={{1,2,3},{2,3,4},{3,4,5},{4,5,6}};

//traversing2Darray

for(i=0;i<4;i++){

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

   printf("arr[%d][%d]=%d\n",i,j,arr[i][j]);

 }//endofj

}//endofi

return 0;

}
```

Output

```
arr[0][0]=1

arr[0][1]=2

arr[0][2]=3

arr[1][0]=2
```

arr[1][1]=3

arr[1][2]=4

arr[2][0]=3

arr[2][1]=4

arr[2][2]=5

arr[3][0]=4

arr[3][1]=5

arr[3][2]=6

MULTIDIMENSIONALARRAYINC

Multidimensionalarraysare one ofthe most powerful featuresoftheC programminglanguage. They allowyou to store data in a ***table-like format***,where each***row***and***column***canbeaccessed usingan***index***.Inthisblogpost,we'lllookatmultidimensionalarraysinC,including their ***syntax***, ***example usage***, and ***output***.

SyntaxofMultidimensionalArraysinC

To create a ***multidimensionalarray*** inC,youneedtospecifythenumberofdimensions and the***size*** of each dimension. The general syntax for declaring a multidimensional array is as follows:

typearray_name[size1][size2]...[sizeN];

Here, ***type***is the***data type***of the elements that will be stored in the array, ***array_name***is the name of the array, and ***size1***, ***size2***, ..., ***sizeN*** arethe sizes of each dimension ofthearray.

For example, the following code declares a ***2-dimensional array***of integers with***3 rows***and***4 columns*:**

intmy_array[3][4];

Itcreates an arraywith***3rows***and***4columns***,foratotalof***12elements***. Eachelementisof type ***int***.

ACCESSINGELEMENTSOFMULTIDIMENSIONALARRAYS

To access an element of a ***multidimensional array***, you need to specify the***indices*** for each dimension.For example,toaccessthe elementinthe ***secondrow***and***third column***of***my_array***, you would use the following syntax:

```
intelement= my_array[1][2];
```

Note that the***indices*** start at***0***, so the first row is***my_array[0]***, the second row is***my_array[1]***, and so on. Similarly, the first column of eachrowis ***my_array[i][0]***, and so on.

INITIALIZINGMULTIDIMENSIONALARRAYS

You caninitializea multidimensionalarraywhenyou declareitby specifyingthevaluesforeach element in the array. For example, the following code declares and initializes a ***2-dimensional array*** of integers with ***2 rows*** and ***3 columns*:**

```
intmy_array[2][3]={
{1,2,3},
{4,5,6}
                    };
```

Itcreatesanarraywith***2rows***and***3columns***andinitializestheelementsto thespecifiedvalues. #include

```
<stdio.h>

intmain(void)
{
   intx[3][2]={{0,1},{2,3},{4,5 }};
   for(inti = 0; i < 3; i++) { for
      (int j = 0;j<2;j++) {
         printf("Elementatx[%i][%i]:",i,j);
         printf("%d\n",x[i][j]);
      }
   }

   return(0);
}
```

OUTPUT:

```
Elementatx[0][0]:0
Elementatx[0][1]:1
Elementatx[1][0]:2
Elementatx[1][1]:3
Elementatx[2][0]:4
Elementatx[2][1]:5
```