KARNATAK LAW SOCIETY’S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University Belagavi)

**(APPROVED BY AICTE, NEW DELHI)**

****

Seminar Report

**TITLE**

**“TWO-DIMENSIONAL ARRAY”**

*Submitted in the partial fulfilment for the academic requirement of*

**1st Semester B.E in**

**PROBLEM SOLVING WITH C**

Submitted by

|  |  |
| --- | --- |
| **NAME OF THE CANDIDATE** | **ROLL NO** |
| **PRAJWAL GHOTAGE** | **I 01** |
| **NAVJEET SINGH KOHLI** | **I 06** |
| **SAHIL HALGEKAR** | **I 07** |
| **HARSHEL MALAWADE** | **I 36** |

**GUIDED BY: R. S. Patil Sir**

**Course Seminar report and ppt content**

**Marks’s allocation:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Batch No.: 08 | | | | | |
| 1. | Seminar Title: | Marks Range | USN | | | |
|  |  |  |  |
| 2. | Abstract (PO2) | 0-2 |  |  |  |  |
| 3. | Application of the topic to the course (PO2) | 0-3 |  |  |  |  |
| 4. | Literature survey and its findings (PO2) | 0-4 |  |  |  |  |
| 5. | Methodology, Results and Conclusion (PO1, PO3,PO4) | 0-6 |  |  |  |  |
| 6. | Report and Oral presentation skill (PO9, PO10) | 0-5 |  |  |  |  |
|  | Total | 20 |  |  |  |  |

**TABLE OF CONTENTS**

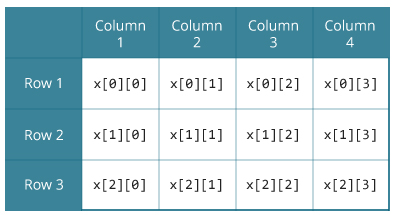
|  |  |
| --- | --- |
|  | Page number |
| 1. Abstract | 4 |
| 1. Introduction | 5 |
| 1. Memory representation of two-dimensional array | 6 |
| 1. Initialization of array and types | 7-8 |
| 1. Accessing the elements of array | 8-9 |
| 1. Passing of two-dimensional array to function | 9 |
| 1. Programs on two-dimensional array | 9-11 |
| 1. Advantages and disadvantages of two-dimensional array | 11 |
| 1. Conclusion | 12 |
| 1. References | 12 |

**ABSTRACT**

This is a seminar report made on the topic two-dimensional array. In this we have explained about the basic concepts of two-dimensional array that is declaration of two-dimensional array, compile time initialization and its types, run time initialization, how a two-dimensional array is stored in memory. Further we have explained how to access the elements of a two-dimensional array, how to pass array to a function. Next, we have discussed a few programs on two-dimensional array, advantages and disadvantages of two-dimensional array. Thus, this report covers most of the topics related to two-dimensional array

**INTRODUCTION**

A **two**-**dimensional array in C** can be thought of as a **matrix** with rows and columns. It has two subscripts one for row and the other for column. First subscript represents the number of rows and second subscript represents the number of columns.



Declaration of two-dimensional array

Syntax: data\_type array\_name[row size][column size];

Here data type specifies the type of data that can be stored in array. Array name can be any valid name which follows all the rules of declaration of a variable. Row size and column size represent the number of rows and columns in the array respectively.

Example: int a[3][3];

Column size

Array name

Row size

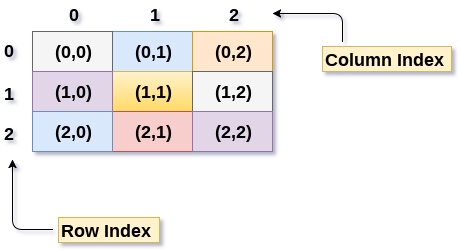
Data type

* Memory representation of two-dimensional array

There are two main techniques of storing 2D array elements into memory

1. Row Major ordering

* In row major ordering, all the rows of the 2D array are stored into the memory contiguously.
* Considering the array shown in the image, its memory allocation according to row major order is shown as follows.

DS 2D Array

first, the 1st row of the array is stored into the memory completely, then the 2nd row of the array is stored into the memory completely and so on till the last row.

* Column Major ordering

According to the column major ordering, all the columns of the 2D array are stored into the memory contiguously. The memory allocation of the array which is shown in in the above image is given as follows.

DS 2D Array

first, the 1st column of the array is stored into the memory completely, then the 2nd row of the array is stored into the memory completely and so on till the last column of the array.

* Initialization of two-dimensional array

Syntax: data\_type array\_name[row size][column size] ={v1,v2,v3…vn};

Example: int a[2][2]={1,2,3,4};

Types of initialization.

1)Initialization of all specified memory locations.

In this type number of elements of array is equal to the size of the array.

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

* 2)Partial initialization

In this type the number of array elements are less than the size of the array.

Example: int a[2][2]={1,3,4};

In this, remaining locations are filled with 0.

3)Initialization without size

In this type either the row size or the column size is not mentioned. The compiler will allocate the memory required on the basis of the input.

Example: int a[2][ ]={{1,2},{3,4}};

In this example the compiler will allocate 2 as the column size.

4)Initialization of the array with elements more than the size of the array

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

This type of initialization will give an error as the number of elements are more than the specified locations, so memory won’t be allocated for those elements which gives error.

These all were types of compile time initialization, that is we cannot change the values present inside the array during the time of execution of the program.

Run time initialization

We can initialize the elements of array during run time.

For this we require two for loops.

Syntax: data\_type array\_name[row size][column size];

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

{

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

` {

scanf(“format specifier”,&array\_name[i][j]);

}

}

Here m is the row size and n is the column size.

* Accessing the elements of two-dimensional array

Consider the two-dimensional array

int a[2][2]={1,2,3,4};

If we want to access an element, we can directly access it by using printf statement.

Syntax: printf(“format specifier”,a[index1][index2];

Example: printf(“%d”,a[0][1]);

This will print 2 on the output screen.

To display or print all the elements of two-dimensional array we again need two for loops.

Syntax: data\_type array\_name[row size][column size];

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

{

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

` {

printf(“format specifier”,&array\_name[i][j]);

}

}

Here m is the row size and n is the column size.

* Passing of two-dimensional array to a function

Syntax

return\_type function\_name(data\_type [ ][n],int,int);

main()

{

local variable declaration;

function\_name(array\_name,row size,column size);

}

return\_type function\_name(data\_type array\_name[ ][n],int row size,int column size);

{

local variable declaration and executable part;

}

* Programs on two-dimensional array

1)Searching an element in two-dimensional array

* Program code

{

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

{

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

{

if(a[i][j]=key)

{

printf(“Element found at location=”);

printf(“%d%d\n”,i,j);

count++;

}

}

}

if(count==0)

printf(“Element not found”);

}

2) C program to find sum of column elements

Code snippet

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

{

sum=0;

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

{

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

}

printf(“Sum of elements of column%d=%d\n”,j+1,sum)

}

3) Multiplication of two arrays

Code snippet

for(i=0;i<r2;i++)

{

for(j=0;j<c2;j++)

{

c[i][j]=0;

for(k=0;k<c2;k++)

{

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

}

}

4) Sum of minor and major diagonal elements

Code snippet

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

{

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

{

if(i+j=m-1)

msum=msum+a[i][j];

if(i==j)

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

}

* Advantages

1)It is used to represent multiple data items of same type by using only one single name.

2)Two-dimensional array are used to represent matrices.

Disadvantages

1)We must know in advance how many elements are to be stored

2)It is static in structure. This means array is of fixed size. The memory which is allocated to array cannot be increased neither be reduced.

3)Since array is fixed size, if we allocate more memory, memory will be wasted, if we allocate less memory, it will create a problem.

4)The elements are stored in consecutive memory locations, so insertion and deletion are difficult and time consuming.

* Applications

1) Used to represent a matrix in programming language, using two-dimensional array we can perform matrix operations like addition, subtraction, transpose, multiplication etc.

2) Used to develop games like tic-tac-toe, solve the n-queens problem etc.

**Conclusion**

Two-dimensional array is one of the major concepts in programming which is used to represent a matrix in programming. By using two-dimensional array, we can do many matrix operations like addition, subtraction, multiplication, transpose etc. Using two-dimensional array, we can develop the tic tac toe game, it is used to solve the n -queens’ problems etc. Thus, these all things make it necessary to learn the basics of two-dimensional array.

**References**

* E.Balaguruswamy “Programming in ANSI C” Tata McGraw hill 5th edition 2010
* “Let us C” by Yashwant Kanetkar