

2D-ARRAYS

Basic 2D-array printing

```
import java.util.*;
public class basicrepresentation {
    public static void main(String args[]){
        int matrix[][]=new int [3][3]; //2d array creation
        Scanner sc = new Scanner(System.in);
        int n=3;
        int m=3;
        for(int i=0;i<n;i++){ // rows
            for(int j=0;j<m;j++){ //columns
                matrix[i][j]=sc.nextInt(); // input stored in matrix
            }
        }
        // printing output
        for(int i=0;i<n;i++){
            for(int j=0;j<m;j++){
                System.out.print(matrix[i][j]+ " ");
            }
            System.out.println(); // new line in printing output
        }
    }
}
```

Output:

```
1 2 3 4 5 6 7 8 9
1 2 3
4 5 6
7 8 9
```

Searching in 2D-array

```
import java.util.*;
public class searchingin2Darray {
public static void Searchinmatrix(int arr[][],int key){
    for(int i=0;i<arr.length;i++){
        for(int j=0;j<arr[0].length;j++){
            if(arr[i][j]==key){
                System.out.println("found at index (" +i+ " "+j+"");
            }
        }
    }
}

public static void main(String args[]){
    int arr[][]=new int[3][3];
    Scanner sc= new Scanner(System.in);

    //input
    for(int i=0;i<arr.length;i++){//rows    n=3 == arr.length
        for(int j=0;j<arr[0].length;j++){//columns    m=3 == arr[0].length
            arr[i][j]=sc.nextInt();
        }
    }

    //output
    for(int i=0;i<arr.length;i++){//rows    n=3 == arr.length
        for(int j=0;j<arr[0].length;j++){//columns    m=3 == arr[0].length
            System.out.print(arr[i][j]+" ");
        }
        System.out.println();// new space to devide out matrix into rows and
columns

    }
    System.out.println("enter key");
    int key=sc.nextInt();
    Searchinmatrix( arr, key);

}
```

```
}
```

Output:

```
1 2 3 4 5 6 7 8 9
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

Enter key

```
5
```

```
found at index (1 1)
```

printing minimum and maximum in 2D-array

```
public class minandmaxin2darray {
    public static int mininarray(int arr[][]){
        int min_value=Integer.MAX_VALUE;//=value(200)
        for(int i=0;i<arr.length;i++){
            for(int j=0;j<arr[0].length;j++){
                if( min_value > arr[i][j]){ //200>1
                    min_value=arr[i][j];// min=1
                }
            }
        }
        return min_value;
    }
    public static void main(String args[]){
        int arr[][]={ {2,4,3},
                      {4,5,6},
                      {7,8,9} };

        System.out.println( mininarray(arr));
    }
}
```

Output:2

```

public class minandmaxin2darray {
    public static int maxinarray(int arr[][]){
        int max_value=Integer.MIN_VALUE;//value(-200)
        for(int i=0;i<arr.length;i++){
            for(int j=0;j<arr[0].length;j++){
                if( max_value < arr[i][j]){ //-200<1
                    max_value=arr[i][j];// max=1
                }
            }
        }
        return max_value;
    }
    public static void main(String args[]){
        int arr[][]={ {2,4,3},
                       {4,5,6},
                       {7,8,9} };

        System.out.println( maxinarray(arr));
    }
}

```

Output:9

Spiral matrix

```

public class spiralmatrix {
    public static void spiralMatrix(int arr[][]){

        int sr=0;
        int sc=0;
        int er=arr.length-1;
        int ec=arr[0].length-1;
        while(sr<=er&&sc<=ec){
            //top
            for(int j=sc;j<=ec;j++){
                System.out.print(arr[sr][j]+" ");
            }
            //right
            for(int i=sr+1;i<=er;i++){
                System.out.print(arr[i][ec]+" ");
            }
        }
    }
}

```

```

    }
    //bottom
    for(int j=ec-1;j>=sc;j--){
        if(sr==ec){
            break;
        }
        System.out.print(arr[er][j]+" ");
    }
    //left
    for(int i=er-1;i>=sr+1;i--){
        if(sc==er){
            break;
        }
        System.out.print(arr[i][sc]+" ");
    }
    sr++;
    sc++;
    er--;
    ec--;

}}
public static void main(String args[]){
    int arr[][]={ {1,2,3,4},
                  {5,6,7,8},
                  {9,10,11,12},
                  {13,14,15,16}};

    spiralMatrix(arr);
}
}

```

Output:

1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

Diagonal Sum

```

public class diagonalsum {
    public static int printdiagsum(int matrix[][]){
        int sum=0;
        //BRUTEFORCE ATTACK   TIME COMPLEXITY=O(n^2)

        // for(int i=0;i<matrix.length;i++){

```

```

        //      for(int j=0;j<matrix[0].length;j++){
        //          if(i==j){
        //              sum+=matrix[i][j];
        //          }
        //          else if(i+j==matrix.length-1){
        //              sum+=matrix[i][j];
        //          }
        //      }
        //  }
        // }

//OPTIMAL SOLLUTION
for(int i=0;i<matrix.length;i++){
    sum+=matrix[i][i]; //primary diagonal
    if(i!=matrix[0].length){
        sum+=matrix[i][matrix.length-1-i];
    }
}
return sum;
}

public static void main(String args[]){
    int matrix[][]={{1,2,3,4},
                    {5,6,7,8},
                    {9,10,11,12},
                    {13,14,15,16}};
    System.out.println( "sum = "+printdiagsum(matrix));
}

```

}
 Output:
 Sum:68

Search in sorted matrix:

```

public class Searchinmatrix {
    //STAIRCASE SEARCH(TOP RIGHT APROACH)

    public static boolean SEARCH(int matrix[][],int key){
        int rows=0;
    }
}

```

```

        int cols=matrix[0].length-1;
        while(rows<=matrix.length && cols>=0){
            if(matrix[rows][cols]==key){
                System.out.println("found at "+rows+", "+cols);
                return true; }
            else if(key>matrix[rows][cols]){
                rows++;
            }
            else{
                cols--;
            }
        }
        return false;
    }
}

public static void main(String args[]){
    int matrix[][]={{1,2,3},
                    {4,5,6},
                    {7,8,9}};

    int key=5;
    System.out.println( SEARCH(matrix,key));
}

```

}

Output:

found at 1,1

true

```

public class Searchinmatrix {
    //STAIRCASE SEARCH (BOTTOMLEFTAPPROACH)

    public static boolean SEARCH(int matrix[][],int key){
        int rows=matrix.length-1;;
        int cols=0;
        while(rows<=matrix.length-1 && cols>=0){
            if(matrix[rows][cols]==key){
                System.out.println("found at "+rows+", "+cols);
                return true; }
            else if(key>matrix[rows][cols]){
                cols++;
            }
            else{
                rows--;
            }
        }
    }
}

```

```

        return false;
    }
    public static void main(String args[]){
        int matrix[][]={{1,2,3},
                        {4,5,6},
                        {7,8,9}};

        int key=5;
        System.out.println( SEARCH(matrix,key));
    }
}

```

}

Output:

found at 1,1

true

TIMECOMPLEXITY =O(N*M)

```

public class assq1 {
    //print the number of digits repeated
    public static void countof3(int matrix[][], int key){
        int count=0;
        for(int i=0;i<matrix.length;i++){

            for(int j=0;j<matrix[0].length;j++){
                if(matrix[i][j]==key){
                    count++;
                }
            }
        }
        System.out.println(count);
    }
}

```

```

public static void main(String args[]){
    int matrix[][]={{1,3,3},
                    {4,5,3}};

    int key=3;

    countof3(matrix, key);
}

}

```

Output: 3

```

public class assq2 {
    //Print out the sum of the numbers in the second row of the "nums" array

    public static void secondrowsum(int matrix[][]){
        int sum=0;
        for(int i=0;i<matrix.length;i++){
            if(i==1){
                for(int j=0;j<matrix[0].length;j++){
                    sum+=matrix[i][j];
                }
                System.out.print("sum of 2nd row:"+sum);
            }
        }

    }

    public static void main(String args[]){
        int matrix[][]= {{1,4,9},
                        {11,4,3},
                        {2,2,3} };
        secondrowsum(matrix);
    }
}

```

Output: sum of 2nd row:18

Transpose of a matrix

```

public class assq3 {
    //Write a program to FindTransposeofa Matrix

    public static void Printarray(int matrix[][]){
        for(int i=0;i<matrix.length;i++){
            for(int j=0;j<matrix[0].length;j++){

                System.out.print(matrix[i][j]+" ");
            }
            System.out.println();
        }

    }
}

```

```
public static void main(String args[]){  
    int rows=2,cols=3;  
    //original array  
    int matrix[][]= {{1,2,3},{4,5,6}};  
    //transpose array  
    int Transpose[][]= new int[cols][rows];  
    for(int i=0;i<rows;i++){  
        for(int j=0;j<cols;j++){  
            Transpose[j][i]=matrix[i][j];  
        }  
    }  
    Printarray(matrix);  
    Printarray(Transpose);  
}
```

}

Output:

```
1 2 3  
4 5 6
```

```
1 4  
2 5  
3 6
```