

Exercise 1 - Sort

Write a program to read an integer of randomly in a 1D array and character array using the random integer.

Apply the method to sort the array content and return the number of comparisons done.

Apply method to Print the sorted array with array index position

Enter the number of elements: 6

| 0 | 1 | 2 | 3 | 4 | 5 |

| 67 | 72 | 74 | 87 | 90 | 97 |

| C | H | J | W | Z | a |

Hint: System.out.printf

random.nextBoolean()true to read for A-Z false for a-z

// Random number in the range [65, 90] for 'A'-'Z' randomNumber = 65 +

random.nextInt(26);

// Random number in the range [97, 122] for 'a'-'z' randomNumber = 97 +

random.nextInt(26);

(char) randomNumber; // Convert integer to character

Hint: Use a method to print the horizontal line based on the size n

code:

```
import java.util.*;
class Sort3003{
    static void line(ArrayList<Integer> l){
        String a="- ";
        for(int i=0;i<l.size();i++){
            System.out.print(a.repeat(6));
        }
        System.out.println();
    }
    static void printl(ArrayList<Integer> l){
        System.out.print("|");
        for(int i:l){
            System.out.printf(" %3d |",i);
        }
        System.out.println();
    }
    static void printc(ArrayList<Integer> l){
        System.out.print("|");
        for(int i:l){
            System.out.printf(" %3c |",i);
        }
        System.out.println();
    }
}
```

```

static void printi(ArrayList<Integer> l,int n){
    System.out.print("|");
    for(int i=0;i<n;i++){
        System.out.printf(" %3d |",i);
    }
    System.out.println();
}
public static void main(String[] args){
    System.out.println("R.Prabhakara Arjun\n2022503003");
    System.out.print("Enter a number:");
    Scanner input=new Scanner(System.in);
    Random r=new Random();
    int n=input.nextInt();
    ArrayList<Integer> l=new ArrayList<>();
    for (int i=0;i<n;i++){
        if(r.nextBoolean()){
            l.add(65+r.nextInt(26));
        }
        else{
            l.add(97+r.nextInt(26));
        }
    }
    //System.out.println(l);
    Collections.sort(l);
    //System.out.println(l);
    line(l);
    printi(l,n);
    line(l);
    printl(l);
    line(l);
    printc(l);
    line(l);
}
}

```

```

PS C:\Users\2022503003\JAVA\WEEK-3> javac Sort3003.java
PS C:\Users\2022503003\JAVA\WEEK-3> java Sort3003
R.Prabhakara Arjun
2022503003
Enter a number:7
-----
|  0 |  1 |  2 |  3 |  4 |  5 |  6 |
-----
| 71 | 79 | 104 | 105 | 109 | 113 | 115 |
-----
|  G |  O |  h |  i |  m |  q |  s |
-----

```

Exercise 2: Search element Occurrence

Write a program to read n random integer in a 1D array of A and B of size n. Apply method to search the occurrence of element in B and print the number of B element occurrence in A .

code:

```
import java.util.*;
class AandB3003{
    public static void main(String[] args){
        System.out.println("R.Prabhakara Arjun\n2022503003");
        System.out.print("Enter a number:");
        Scanner input=new Scanner(System.in);
        Random r=new Random();
        int n=input.nextInt();
        ArrayList<Integer> l=new ArrayList<>();
        ArrayList<Integer> m=new ArrayList<>();
        for (int i=0;i<n;i++){
            l.add(r.nextInt(10));
        }
        System.out.println(l);
        for (int i=0;i<n;i++){
            m.add(r.nextInt(10));
        }
        System.out.println(m);
        HashMap<Integer,Integer> mapp=new HashMap<>();//put,get,containskey
        for(int i:l){
            if(mapp.containsKey(i)){
                mapp.put(i,mapp.get(i)+1);
            }
            else{
                mapp.put(i,1);
            }
        }
        System.out.println(mapp);
        HashMap<Integer,Boolean> mappi=new HashMap<>();
        for(int i:m){
            mappi.put(i,true);
        }
        System.out.println(mappi);
        for(int i:m){
            //System.out.println(mappi);
            if(mapp.containsKey(i) && mappi.get(i)){
                System.out.println(i+" from B is present "+mapp.get(i)+" times in A");
                mappi.put(i,false);
            }
            else if(mappi.get(i)){
                System.out.println(i+" from B is NOT present in A");
                mappi.put(i,false);
            }
        }
    }
}
```

```

    }

}
}
PS C:\Users\2022503003\JAVA\WEEK-3> javac AandB3003.java
PS C:\Users\2022503003\JAVA\WEEK-3> java AandB3003
R.Prabhakara Arjun
2022503003
Enter a number:20
[0, 2, 0, 7, 5, 4, 8, 4, 6, 7, 4, 2, 7, 1, 3, 4, 4, 1, 2, 2]
[6, 2, 8, 6, 0, 7, 9, 6, 0, 7, 0, 5, 1, 6, 1, 3, 9, 5, 5, 0]
6 from B is present 1 times in A
2 from B is present 4 times in A
8 from B is present 1 times in A
0 from B is present 2 times in A
7 from B is present 3 times in A
9 from B is NOT present in A
5 from B is present 1 times in A
1 from B is present 2 times in A
3 from B is present 1 times in A
PS C:\Users\2022503003\JAVA\WEEK-3>

```

Exercise 3: Sum of arrays

Write a program to read two 2D array. Apply method to perform column major sum and sort the array based on the sum of columns.
Sample Input:

1	3	4		2	1	3		3	4	7		4	3	7
1	2	4		5	-1	2	=>	6	1	6	=>	1	6	6
3	1	2	+	3	4	2		6	5	4		5	6	4

Code:

```

import java.util.*;
class matrix{
    ArrayList<ArrayList<Integer>> m=new ArrayList<>();
    Random ri=new Random();
    matrix(int r,int c){
        for(int i=0;i<r;i++){
            ArrayList<Integer> row=new ArrayList<>();
            for(int j=0;j<c;j++){
                row.add(ri.nextInt(10));
            }
            m.add(row);
        }
        //System.out.println(m);
        print(m);
    }
    matrix(int r){

```

```

        for(int i=0;i<r;i++){
            ArrayList<Integer> row=new ArrayList<>();
            m.add(row);
        }
        System.out.println(m);
    }
    void print(ArrayList<ArrayList<Integer>> m){
        for(int i=0;i<m.size();i++){
            for(int j=0;j<m.get(i).size();j++){
                System.out.printf("%d ",m.get(i).get(j));
            }
            System.out.println();
        }
    }
    ArrayList<ArrayList<Integer>> transpose(ArrayList<ArrayList<Integer>> x){
        if (x.isEmpty() || x.get(0).isEmpty()) {
            return new ArrayList<>();
        }
        for(int i=0;i<x.size();i++){
            for(int j=0;j<i;j++){
                int temp=x.get(i).get(j);
                x.get(i).set(j,x.get(j).get(i));
                x.get(j).set(i,temp);
            }
        }
        return x;
    }
    int sum(ArrayList<Integer> list) {
        int total = 0;
        for (int num : list) {
            total += num;
        }
        return total;
    }
    void sumi(ArrayList<ArrayList<Integer>> a,ArrayList<ArrayList<Integer>> b){
        a=transpose(a);
        b=transpose(b);
        print(a);
        System.out.println();
        print(b);
        ArrayList<ArrayList<Integer>> c=new ArrayList<>();
        for(int i=0;i<a.size();i++){
            ArrayList<Integer> row=new ArrayList<>();
            c.add(row);
        }
        for(int i=0;i<a.size();i++){
            for(int j=0;j<a.get(i).size();j++){
                c.get(i).add((a.get(i).get(j)+b.get(i).get(j)));
            }
        }
    }

```

```

    }
}
System.out.println();
print(c);
ArrayList<Integer> columnSums = new ArrayList<>();
ArrayList<Integer> indices = new ArrayList<>();

for (int i = 0; i < c.size(); i++) {
    columnSums.add(sum(c.get(i)));
    indices.add(i);
}

for (int i = 0; i < columnSums.size() - 1; i++) {
    for (int j = i + 1; j < columnSums.size(); j++) {
        if (columnSums.get(i) > columnSums.get(j)) {
            int tempSum = columnSums.get(i);
            columnSums.set(i, columnSums.get(j));
            columnSums.set(j, tempSum);

            int tempIndex = indices.get(i);
            indices.set(i, indices.get(j));
            indices.set(j, tempIndex);
        }
    }
}
ArrayList<ArrayList<Integer>> sortedMatrix = new ArrayList<>();
for (int index : indices) {
    sortedMatrix.add(c.get(index));
}
System.out.println();
print(sortedMatrix);
sortedMatrix=transpose(sortedMatrix);
System.out.println();
print(sortedMatrix);
}
}
class matrix3003{
    public static void main(String[] args){
        Random r=new Random();
        int row=2+r.nextInt(5);
        int col=row;
        matrix a=new matrix(row,col);
        System.out.println();
        matrix b=new matrix(row,col);
        matrix c=new matrix(row);
        c.sumi(a.m,b.m);
    }
}

```

```

PS C:\Users\DELL\OneDrive\Desktop\MIT\CODES\FOR GIT HUB\JAVA_BASICS\WEEKLY LAB_EXERCISE\week-3> java matrix3003.java
1 8
7 5

7 4
5 6
[[[]], [[]]
1 7
8 5

7 5
4 6

8 12
12 11

8 12
12 11

8 12
12 11

```

Exercise 4: Magic square

Write a magic square game using 2darray such that All the row, column and diagonal sum are equal. Create a 3 X 3 grid using array of the integers 1 to n^2 . Read the random integer for the middle position and then fill the other places to obtain magic square.

CODE:

```

import java.util.*;
class magic3003{
    public static void main(String[] args){
        Scanner input=new Scanner(System.in);
        System.out.println("R.Prabhakara Arjun\n2022503003");
        System.out.print("Enter a odd number(ROWS*COLS,n(Rows)=n(cols)) to generate a magic square:");
        int n=input.nextInt();
        if (n%2==0){
            System.out.println("You just can't understand what I said right!\nRun the program again!!");
            return;
        }
        int[][] magicsq=genMagicSq(n);
        for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                System.out.printf("%4d",magicsq[i][j]);
            }
            System.out.println();
        }
    }
    public static int[][] genMagicSq(int n){
        int[][] magicsq=new int[n][n];
        int num=1;
        int row=0;
        int col=n/2;
        while(num<=n*n){ //3-->9(1,2,3,4,5,6,7,8,9)
            magicsq[row][col]=num++;

```

```

        int newr=(row+n-1)%n;
        int newc=(col+1)%n;
        if(magicsq[newr][newc]!=0){
            row=(row+1)%n;
        }
        else{
            row=newr;
            col=newc;
        }
    }
    return magicsq;
}
}

```

```

● PS C:\Users\DELL\OneDrive\Desktop\MIT\CODES\FOR GIT HUB\JAVA_BASICS\WEEKLY_LAB_EXERCISE\week-3> java magic3003.java
R.Prabhakara Arjun
2022503003
Enter a odd number(ROWS*COLS,n(Rows)=n(cols)) to generate a magic square:5
17 24 1 8 15
23 5 7 14 16
4 6 13 20 22
10 12 19 21 3
11 18 25 2 9
● PS C:\Users\DELL\OneDrive\Desktop\MIT\CODES\FOR GIT HUB\JAVA_BASICS\WEEKLY_LAB_EXERCISE\week-3> java magic3003.java
R.Prabhakara Arjun
2022503003
Enter a odd number(ROWS*COLS,n(Rows)=n(cols)) to generate a magic square:7
30 39 48 1 10 19 28
38 47 7 9 18 27 29
46 6 8 17 26 35 37
5 14 16 25 34 36 45
13 15 24 33 42 44 4
21 23 32 41 43 3 12
22 31 40 49 2 11 20
● PS C:\Users\DELL\OneDrive\Desktop\MIT\CODES\FOR GIT HUB\JAVA_BASICS\WEEKLY_LAB_EXERCISE\week-3> java magic3003.java
R.Prabhakara Arjun
2022503003
Enter a odd number(ROWS*COLS,n(Rows)=n(cols)) to generate a magic square:11
68 81 94 107 120 1 14 27 40 53 66
80 93 106 119 11 13 26 39 52 65 67
92 105 118 10 12 25 38 51 64 77 79
104 117 9 22 24 37 50 63 76 78 91
116 8 21 23 36 49 62 75 88 90 103
7 20 33 35 48 61 74 87 89 102 115
19 32 34 47 60 73 86 99 101 114 6
31 44 46 59 72 85 98 100 113 5 18
43 45 58 71 84 97 110 112 4 17 30
55 57 70 83 96 109 111 3 16 29 42
56 69 82 95 108 121 2 15 28 41 54
○ PS C:\Users\DELL\OneDrive\Desktop\MIT\CODES\FOR GIT HUB\JAVA_BASICS\WEEKLY_LAB_EXERCISE\week-3>

```