DSA LAB 1

Azhar Ali
023-23-0314
Sec A

- Write a function in Java (or any language) to read a list of 10 integer numbers and arrange them in such a manner that all the even numbers start from the left and all the odd numbers start from the right.

  Input: 1 2 3 5 7 2 2 7 8 9

  Output: 1 3 5 7 7 9 2 2 2 8

```java
class Qno1{

    public static void Arrange(int a []){
        int tempod=0;
        int tempev = 0;

        int odd[] = new int[a.length];
        int even[] = new int[a.length];

        for(int i=0 ; i<a.length; i++){
            if (a[i]%2==0){
                even[tempev++]=a[i];
            }
            else if (a[i]%2==1){
                odd[tempod++]=a[i];
            }
        }
        for (int b=0;b<odd.length ; b++)
        {   if(odd[b]==0){
            continue;
        }
            System.out.print(odd[b]+" ");
            }
        for (int c=0;c<even.length ; c++)
        {   if(even[c]==0){
            continue;
        }
            System.out.print(even[c]+" ");
        }}}
class Main{
    Run | Debug
    public static void main(String[] args) {

        Qno1 test = new Qno1();
        int Array[]={1,2 ,3 ,5, 7 ,2 ,2, 7, 8, 9};
        System.err.println("input");
        for (int i : Array) {
            System.out.print(i+" ");
        }
        System.out.println("\noutput");
        test.Arrange(Array);
      }
}
```

```
input
1 2 3 5 7 2 2 7 8 9
output
1 3 5 7 7 9 2 2 2 8
PS D:\Semester 3\DSA LAB>
```

2. Write a function named noDup() that takes a 2D array of size 4x5 and a 1D array of size 20. It should then copy all the elements of 2D array into 1D array but should avoid duplication.

Qno2.java > Qno2 > noDp(int[][], int[])

```java
public class Qno2 {
    void noDp(int a[][],int b[]){
        int z=0;
        int num;
        for (int row = 0 ; row<4 ; row++){
            for(int col=0 ; col<5 ; col++ ){
                num=a[row][col];
                //  System.out.println(num);
                boolean isdup=false;
                for(int i = 0 ;i<20 ;i++){
                    if (num==b[i]) {
                        isdup=true;
                        break;
                    }
                }
                if (!isdup) {

                        b[z++] = num;
                }
        } }}
    Run | Debug
    public static void main(String[] args) {
        int arr2d[][] = {{1,3,5,7,9},{2,4,6,8,2},{2,3,4,5,6},{3,4,5,6,7,}};
        int arr1d[]=new int[20];

        Qno2 test = new Qno2();
        System.out.println("From 2D array");
        for(int i = 0 ; i <arr2d.length ; i++){
            for(int j = 0 ; j<arr2d[i].length ; j++){
                System.out.print(arr2d[i][j]+"  ");
            }
            System.out.println();
        }
        test.noDp(arr2d, arr1d);
        System.out.println("\nInto Single");
        for(int i=0;i<20;i++){
            if(arr1d[i]==0){
                continue;
            }
            else
            System.out.print(arr1d[i]+" ");
        }
    }
}
```

```
b85d80c849f0a30b5e5deacd48d952
From 2D array
1  3  5  7  9
2  4  6  8  2
2  3  4  5  6
3  4  5  6  7

Into Single
1 3 5 7 9 2 4 6 8
PS D:\Semester 3\DSA LAB>
```

3. Create a file named NLArray.java and design following functions or performing NLP. -

- String [] wordTokenize (String fileName) ◊ Read any text file and return list of words from that file. (Ignore . , : and all these types operators)

```
String WordTokenize(String A)
{
      String re="";
       for(int i=0;i<A.length() ; i++)
       {
            if (A.charAt(i)=='.' || A.charAt(i)==':' || A.charAt(i)==',' || A.charAt(i)==';')
                {
                   continue;
                }
            else {
                 re += A.charAt(i);
                }
       }
          return re;
   }
```

4. Design following methods in above same class NLArray.java for Image Cropping. - -

- void extractBoundaries (int arr[][]) ◊ This function should extract boundaries and print from arr (Boundaries include 1st row, 1st col, last row, last col).

```
// }
void ExtractBoundries(int arr[][]){
    for(int row = 0 ; row < arr.length ; row++){
        for (int col = 0 ; col<arr[row].length ; col++) {
            if(row == 0 || row==(arr.length-1) || col ==0 || col==(arr[0].length)-1){
                System.out.print(arr[row][col]+" ");
            }
            else {
            System.out.print("  ");
            }
        }
        System.out.println();
    }
}
```

- void cropCenterPart (int arr[][]) ◊ This function should extract center part and print from arr, center part includes everything except Boundaries (Boundaries include 1st row, 1st col, last row, last col).

```
void CropCenter(int arr[][]){
    for(int row = 0 ; row < arr.length ; row++){
        for (int col = 0 ; col<arr[row].length ; col++) {
            if(row == 0 || row==(arr.length-1) || col ==0 || col==(arr[0].length)-1){
               System.err.print(" ");
            }
            else {
                System.out.print(arr[row][col]+" ");
            }
        }
        System.out.println();
    }
}
```

s

- **MAIN Method And Output of Qno 3 and Qno 4**

```
Run | Debug
public static void main(String[] args) {
    String test = "Ignore, special:. oprators";
    NLArray nl = new NLArray();
    System.out.println("Before wordtokenize method");
    System.out.println(test+"\n");
    System.out.println("After wordtokenize method");
   System.out.println(nl.WordTokenize(test)+"\n");

   int arr[][] = {{1,2,3,4,5},{1,2,3,4,5},{1,2,3,4,5},{1,2,3,4,5}};
   System.out.println("Before Boundry Extract");
   for(int row = 0 ; row<arr.length ; row++){
    for(int col = 0 ; col<arr[row].length ; col ++){
        System.out.print(arr[row][col]+" ");
    }
    System.out.println();
   }
   System.out.println();

   System.out.println("After apply Boundry Extract");
   nl.ExtractBoundries(arr);

   System.out.println("\nAfter Apply center Method");
   nl.CropCenter(arr);


}
}
```

```
Before wordtokenize method
Ignore, special:. oprators

After wordtokenize method
Ignore special oprators

Before Boundry Extract
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5
1 2 3 4 5

After apply Boundry Extract
1 2 3 4 5
1       5
1       5
1 2 3 4 5

After Apply center Method

 2 3 4
 2 3 4

PS D:\Semester 3\DSA LAB>
```