

أسئلة على الأريه ليست

شعبة 3

محمد ابو صفط

1* ميثود يستقبل اريه لسيت ويعطيني اكبر عنصر موجود فيها :

Input : {1,5,9,7,7}

Output: 9

```
public static int Max(ArrayList<Integer> arr){  
    if(arr.size()==1) //في حال كان عندي عنصر واحد فالمصفوفة  
        return arr.get(0);  
  
    int max = arr.get(0);  
    for (int i = 1; i < arr.size() ; i++) {  
        if(arr.get(i)>max)  
            max=arr.get(i);  
    }  
    return max;  
}
```

2* ميثود يستقبل 2 اريه ليست ويرجع العناصر المشتركة بينهم

input: a={1,7,7,8,9,2,1} / b= {1,3,7,5}

output: {1,7}

```
public static ArrayList<Integer> Common(ArrayList<Integer> a , ArrayList<Integer> b){  
    ArrayList<Integer> arr = new ArrayList<>(); //الاريه اللي بدي ارجعها  
    for (int i = 0; i < a.size(); i++){ //بدنا نمشي على كل عناصر الاريه الاولى واحد ونشوف اذا موجود في الاريه الثانية  
        if(b.contains(a.get(i))){  
            if(arr.contains(a.get(i))==false) //عشان ما نكرر العناصر  
                arr.add(a.get(i));  
        }  
    }  
    return arr;  
}
```

*3 ميثود يستقبل اريه ليست ويعكسها

input: {1,7,7,8,9,2,1}

output: {1,2,9,8,7,7,1}

```
public static ArrayList<Integer> reverse(ArrayList<Integer> arr){  
    for (int i = 0; i < arr.size()/2; i++) {  
        arr.set(i, arr.set(arr.size()-i-1, arr.get(i)));  
    }  
    return arr;  
}
```

طريقة ثانية:-

```
public static ArrayList<Integer> reverse(ArrayList<Integer> arr){  
    int temp;  
    for (int i = 0; i < arr.size()/2; i++) {  
        temp = arr.get(i);  
        arr.set(i, arr.get(arr.size()-i-1));  
        arr.set(arr.size()-i-1, temp);  
    }  
    return arr;  
}
```

*4 ميثود يفحص اذا كانت الاريه بلندروم

بلندروم -- {1,2,3,2,1}

```
public static boolean CheckPalendrom(ArrayList<Integer> arr){  
    for (int i = 0; i < arr.size()/2; i++) {  
        if(arr.get(i) != arr.get(arr.size()-i-1))  
            return false;  
    }  
    return true;  
}
```

*5 ميثود يسوي دوران للشمال بمقدار (ن)

Input : {1,2,3,4,5} , 2

Output : {3,4,5,1,2}

```
public static void Rleft(ArrayList<Integer> arr , int n){  
    for (int i = 0; i < n; i++) {  
        arr.add(arr.get(0));  
        arr.remove(index: 0);  
    }  
}
```

اذا بدنا نسوي الميثود داخل كلاس الاريه ليست اللي انشأناه وهالمرة لليمين

```
Public void Rright(int n){  
    for (int i = 0; i < n; i++) {  
        this.addFirst(this.get(this.size()-1);  
        this.removeLast();  
    }  
}
```

*6 میثود مستقل اریه لیست ویرجع اکبر 3 اعداد

Input: {1,5,6,7,2,7}

Output: {7,7,6}

```
public static int[] Max3(ArrayList<Integer>arr){
    int[] a = new int[3];
    a[0] = a[1] = a[2] = Integer.MIN_VALUE; // اصغر قيمة انتجها فالكلاس (كل الاعداد اكبر منه)
    // a[0] اكبر عدد
    // a[1] ثاني اكبر عدد
    // a[2] ثالث اكبر عدد
    for (int i = 0; i < arr.size() ; i++) {
        if(arr.get(i)>a[0]){
            a[2]=a[1];
            a[1]=a[0];
            a[0]=arr.get(i);
        }
        else if(arr.get(i)>a[1]){
            a[2]=a[1];
            a[1]=arr.get(i);
        }
        else if(arr.get(i)>a[2])
            a[2]=arr.get(i);
    }
    return a;
}
```

إذا بدى ما يرجعلي مكرر بصير لازم ابعثله اريه علاقل لازم يكون 3 عناصر مختلفة فيها

Input: {1,5,6,7,2,7}

Output: {7,6,5}

```
public static int[] Max3(ArrayList<Integer>arr){
    int[] a = new int[3];
    a[0] = a[1] = a[2] = Integer.MIN_VALUE; // اصغر قيمة انتجها فالكلاس (كل الاعداد اكبر منه)
    // a[0] اكبر عدد
    // a[1] ثاني اكبر عدد
    // a[2] ثالث اكبر عدد
    for (int i = 0; i < arr.size() ; i++) {
        if(arr.get(i)==a[0]||arr.get(i)==a[1]||arr.get(i)==a[2])
            continue;

        else if(arr.get(i)>a[0]){
            a[2]=a[1];
            a[1]=a[0];
            a[0]=arr.get(i);
        }
        else if(arr.get(i)>a[1]){
            a[2]=a[1];
            a[1]=arr.get(i);
        }
        else if(arr.get(i)>a[2])
            a[2]=arr.get(i);
    }
    return a;
}
```

*7 ميثود يستقبل اريه ليست ويحذف العناصر المتكررة

Input: {1,5,1,7,2,7}

Output: {1,5,7,2}

```
public static void removeD(ArrayList<Integer> arr){
    for (int i = 0; i < arr.size()-1; i++) { //1- حطينا اذا ما حطينا
        for (int j = i+1; j < arr.size(); j++) {
            if(arr.get(i)==arr.get(j)) {
                arr.remove(j);
                j--; //--j عملنا عنه يفشق عنه بعدد فحشاش ما يفشق عنه عملنا j--
            }
        }
    }
}
```

*8

5) write a method that moves all zeros to the end of the given ArrayList. Examples:

input: {1, 2, 3, 0, 4, 5}

output: {1, 2, 3, 4, 5, 0}

-

input: {0, 0, 0, 1}

output: {1, 0, 0, 0}

```
public static void move0(ArrayList<Integer> arr){
    for (int i = arr.size()-1; i >= 0 ; i--) {
        if(arr.get(i)==0)
            arr.add(arr.remove(i)); //remove(i) 0 هترجع 0
    }
}
```

Given an array list of integers "list" and two numbers "index" , "k". remove K consecutive elements from list starting at index.

method header: `public static void removeConsecutive(ArrayList<Integer>list, int index, int k)`

Example:

input: list = [1,2,3,4,5,6,7,8,9], index = 2, k = 5

after using the method: [1,2,8,9]

```
public static void RemoveConsecutive(ArrayList<Integer> arr , int index ,int k){
    for (int i = 1; i <=k ; i++) { // الفور لوب لعدد مرات الحذف وليس الوصول للعنصر المحذوف
        arr.remove(index);
    }
}
```

6) you are given an array list of pairs. each pair is an integer array with length= 2

in the form of [N, value].return an array list of integers which contains each value N number of times. for example:

input: { [1,2] , [3,4] , [2,3] }

output: { 2, 4, 4, 4, 3, 3 }

method header: `public static ArrayList<Integer> decoded(ArrayList<int[]> incoded)`

```
public static ArrayList<Integer> sum(ArrayList<int[]> arr){
    ArrayList<Integer> arr2 = new ArrayList<>();
    for (int i = 0; i < arr.size(); i++) {
        for (int j = 0; j < arr.get(i)[0]; j++) {
            arr2.add(arr.get(i)[1]);
        }
    }
    return arr2;
}
```


Given an array list that represents people. each person has multiple bank accounts(array list of integers). find the how much money the richest person has.

method header: public static int maxSum(ArrayList<ArrayList<Integer>> people)

Example: input: {{1,2,3,4,5},{18,1},{3,6,4}}

output: 19

```
public static int MaxSum(ArrayList<ArrayList<Integer>> arr){

    int max=0;
    int sum =0;
    for (int i = 0; i < arr.size(); i++) { // رح تمشي على اريه داخل الاريه الكبيره
        for (int j = 0; j < arr.get(i).size(); j++) { // رح تمشي على عناصر الاريه الداخليه
            sum+=arr.get(i).get(j);
        }
        if(max<sum)
            max=sum;
        sum=0;
    }
    return max;
}
```

Given an array list of integers "list" and a number "k". divide this list into k lists of equal length, and return an array list that contains these lists in order.

NOTE: You can assume that list.size() is divisible by k with no remainder.

method header: public static ArrayList<ArrayList<Integer>> divideList(ArrayList<Integer> list, int k)

Example: input: list = {1,2,3,4,5,6,7,8,9}, k = 3

output: {{1,2,3},{4,5,6},{7,8,9}}

```
public static ArrayList<ArrayList<Integer>> DivideList(ArrayList<Integer> arr , int k){

    ArrayList<ArrayList<Integer>> arr2 = new ArrayList<>();
    int c=0;
    for (int i = 0; i < k; i++) {
        arr2.add(new ArrayList<>());
        for (int j = 0; j < arr.size()/k; j++) {
            arr2.get(i).add(arr.get(c++));
        }
    }
    return arr2;
}
```