Programming Assessment - Zoho Corporation PTA

Name: Swathy krishna
Email id: swathykrishna2022b@mca.ajce.in

Q1. <u>Sum Equal Target</u>

Given an array of integers and a target value, determine if there are any two integers in the array whose sum is equal to the given target value. Return true if the sum exists and return false if it does not.

Constraint:

1<=integers<=100

Input:

No. of integers = 4

Numbers = [2, 7, 11, 15]

Target Sum = 9,

Output:

Possible

Input:

No. of integers = 5

Numbers = [2,4,5,7,8]

Target Sum = 11,

Output:

Possible

Input:

No. of integers = 5

Numbers = [2,8,5,7,18]

Target Sum = 3,

Output:

Not Possible

Answer:

Java code:

```
/*Given an array of integers and a target value, determine if there are any two integers in the array
whose sum is equal to the given target value. Return true if the sum exists and return false if it does
not.*/
import java.util.*;
class Main
{
public static void main(String args[])
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter length of the array:");


int len=sc.nextInt();
int arr[]=new int[len];
System.out.println("Enter array elements:");
for(int i=0;i<len;i++)
```

```
 arr[i]=sc.nextInt();
 System.out.println("Enter target value:");
 int target=sc.nextInt();
 int flg=0;
 for(int i=0;i<len;i++)
 {
 for(int j=0;j<len;j++)
 {
 if(i==j)
 continue;
 int sum=arr[i]+arr[j];
 if(sum==target)
 flg=1;
 }
 }
 if(flg==1)
 System.out.println("Possible");
 else
 System.out.println("Not Possible");
 }
}
```

Output

```
Enter length of the array:
4
Enter array elements:
2 7 11 15
Enter target value:
9
Possible
```

```
Enter length of the array:5
Enter array elements:
2 4 5 7 8
Enter target value:
11
Possible
```

```
Enter length of the array:5
Enter array elements:2 8 5 7 18
Enter target value:3
Not Possible
```

Q2. Smashing the Stones

You are given an array of integers stones where stones[i] is the weight of the ith stone.
We are playing a game with the stones. On each turn, we choose the heaviest two stones and smash them together. Suppose the heaviest two stones have weights x and y with x <= y.

The result of this smash is:

- If x == y, both stones are destroyed, and
- If x != y, the stone of weight x is destroyed, and the stone of weight y has new weight y - x.

At the end of the game, there is at most one stone left.
Return the smallest possible weight of the left stone. If there are no stones left, return 0.

Answer

Java code:

```
/* Smashing the Stones
You are given an array of integers stones where stones[i] is the weight of the ith stone.
We are playing a game with the stones. On each turn, we choose the heaviest two stones and smash
them together */

import java.util.*; class Pgm2
{
int arr[],index=0,sindex=0,result[]; Pgm2()
{
Scanner sc=new Scanner(System.in); System.out.println("Enter length of the array:"); int len=sc.nextInt();
arr=new int[len];
System.out.println("Enter array elements:"); for(int i=0;i<len;i++)
arr[i]=sc.nextInt();
}
void largest()

{
int l=arr[0];
for(int i=0;i<arr.length;i++)
{
if(arr[i]>l)
{
l=arr[i]; index=i;
}
}
int sl=0;
for(int i=0;i<arr.length;i++)
{
if(index==i)
continue; if(arr[i]>sl)


{
sl=arr[i]; sindex=i;
}
}
}
void smash()
{
if(arr[index]==arr[sindex])
{
```

```
result=new int[arr.length-2];
for(int i=0,j=0;j<result.length;i++,j++)
{

}
}
else

if(i!=index|| i!=sindex)
result[j]=arr[i];

{

if(arr[index]<arr[sindex])
{
int w=arr[sindex]-arr[index]; result=new int[arr.length-1]; arr[sindex]=w;
for(int i=0,j=0;j<result.length;i++)
{

}
}
else
{

if(i==index)
continue; result[j]=arr[i];j++;

int w=arr[index]-arr[sindex]; result=new int[arr.length-1]; arr[index]=w;
for(int i=0,j=0;j<result.length;i++)
{
if(i==sindex)
continue; result[j]=arr[i];j++;
}
}
}
for(int i=0;i<result.length;i++) System.out.println(result[i]);

arr=new int[result.length]; for(int i=0;i<result.length;i++)
arr[i]=result[i];
}

public static void main(String args[])

{
Pgm2 obj=new Pgm2(); int c=1; while(obj.arr.length>1)
{
System.out.println("SMASH "+c); obj.largest();
obj.smash(); c++;


}

}
}
```

Output

```
Enter length of the array:7
Enter array elements:
1 150 190 170 -1 -1 160 180
SMASH 1
1
150
20
-1
-1
160
SMASH 2
1
20
-1
-1
10
SMASH 3
1
10
-1
-1
```

Q3. <u>Sort Without Moving</u>

Some people are standing in a row in a park. There are trees between them which cannot be moved. Your task is to rearrange the people by their heights in a non-descending order without moving the trees.

constraints:

- Given row r, $5 \leq r.length \leq 15$,
- maximum height of the people could be 200.
- Height of the tree is denoted as -1
- **[output]** Sorted row with all the trees untouched.

Example

For a = [-1, 150, 190, 170, -1, -1, 160, 180], the output should be
sortByHeight(a) = [-1, 150, 160, 170, -1, -1, 180, 190]

Answer

Java code:

```
/* Some people are standing in a row in a park. There are trees between them which
cannot be moved. Your task is to rearrange the people by their heights in a non-
descending order without moving the trees.*/

import java.util.*;
class Main
```

```
 {
 int arr[],index=0,sindex=0,result[];
 Main()
{
 Scanner sc=new Scanner(System.in);
 System.out.println("Enter length of the array:");
 int len=sc.nextInt();
 arr=new int[len];
 System.out.println("Enter array elements:");
 for(int i=0;i<len;i++)
 arr[i]=sc.nextInt();
 }
 void largest()
 {
 int l=arr[0];
 for(int i=0;i<arr.length;i++)
 {
 if(arr[i]>l)
 {
 l=arr[i];
 index=i;
 }
 }
 int sl=0;
 for(int i=0;i<arr.length;i++)
 {
 if(index==i)
 continue;
 if(arr[i]>sl)
 {
 sl=arr[i];
 sindex=i;
 }
 }
 }
 void smash()
 {
 if(arr[index]==arr[sindex])

 {
 result=new int[arr.length-2];
 for(int i=0,j=0;j<result.length;i++,j++)
 {
 if(i!=index|| i!=sindex)
 result[j]=arr[i];
 }


 }
 else
```

```
{
if(arr[index]<arr[sindex])
{
int w=arr[sindex]-arr[index];
result=new int[arr.length-1];
arr[sindex]=w;


for(int i=0,j=0;j<result.length;i++)
{
if(i==index)
continue;
result[j]=arr[i];j++;
}
}
else
{
int w=arr[index]-arr[sindex];
result=new int[arr.length-1];
arr[index]=w;
for(int i=0,j=0;j<result.length;i++)
{
if(i==sindex)
continue;
result[j]=arr[i];j++;
}
}
}
for(int i=0;i<result.length;i++)
System.out.println(result[i]);
arr=new int[result.length];
for(int i=0;i<result.length;i++)
arr[i]=result[i];
}
public static void main(String args[])
{
Main obj=new Main();
int c=1;
while(obj.arr.length>1)
{
System.out.println("SMASH "+c);


obj.largest();
obj.smash();
c++;
}
}
}
```

Output

```
Enter length of the array:7
Enter array elements:
-1 150 190 170 -1 -1 160 180
Sorted list is
-1 150 160 170 -1 -1 190
```