Sukkur IBA University
Department of Computer Science

DATA STRUCTURES
CountDistinctIntegers

Instructor: Saif Hassan

READ IT FIRST

Prior to start solving the problems in this assignments, please give full concentration on following points.

1. WORKING – This is individual lab. If you are stuck in a problem contact your teacher, but, in mean time start doing next question (don't waste time).
2. DEADLINE – 11th March, 2022
3. SUBMISSION – This assignment needs to be submitted in a soft copy.
4. WHERE TO SUBMIT – Please visit your LMS.
5. WHAT TO SUBMIT – Submit this docx and pdf file.

KEEP IT WITH YOU!

1. Indent your code inside the classes and functions. It's a good practice!
2. It is not bad if you keep your code indented inside the loops, if and else blocks as well.
3. Comment your code, where it is necessary.
4. Read the entire question. Don't jump to the formula directly.

I, _**Amjad Ali**_with student ID _**191-21-0001**_

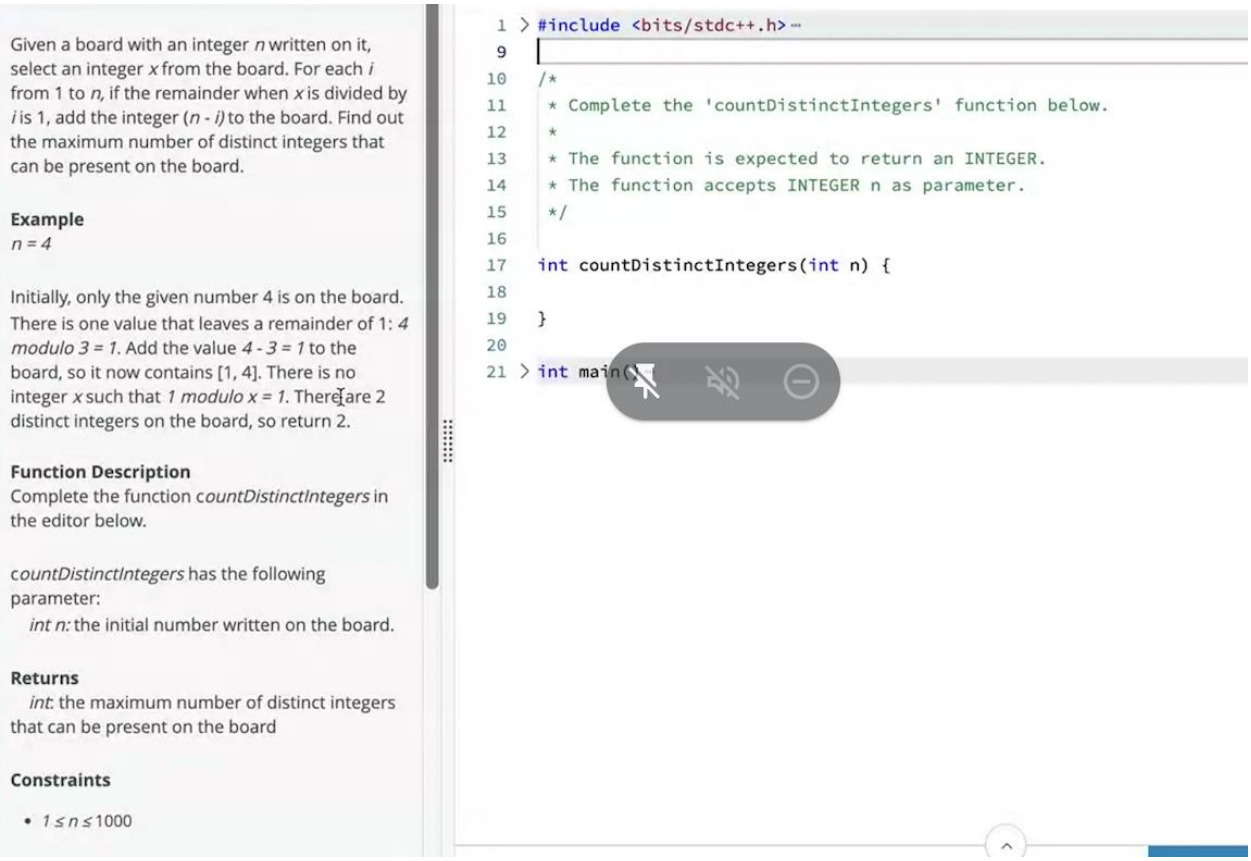
Section _**A**__hereby declare that I do understand the instructions above and follow them. This is

my own work.

Exercises

Task1 Description

…



Given a board with an integer *n* written on it, select an integer *x* from the board. For each *i* from 1 to *n*, if the remainder when *x* is divided by *i* is 1, add the integer (*n - i*) to the board. Find out the maximum number of distinct integers that can be present on the board.

Example
n = 4

Initially, only the given number 4 is on the board. There is one value that leaves a remainder of 1: *4 modulo 3 = 1*. Add the value *4 - 3 = 1* to the board, so it now contains [1, 4]. There is no integer *x* such that *1 modulo x = 1*. There are 2 distinct integers on the board, so return 2.

Function Description
Complete the function *countDistinctIntegers* in the editor below.

countDistinctIntegers has the following parameter:
int n: the initial number written on the board.

Returns
int: the maximum number of distinct integers that can be present on the board

Constraints

- $1 \le n \le 1000$

```
1  #include <bits/stdc++.h>
9
10 /*
11  * Complete the 'countDistinctIntegers' function below.
12  *
13  * The function is expected to return an INTEGER.
14  * The function accepts INTEGER n as parameter.
15  */
16
17 int countDistinctIntegers(int n) {
18
19 }
20
21 int main()
```

Solution:

```
1. static HashSet<Integer> set=new HashSet<>();
2.     public static int numberOfDistinctIntegers(int n){
3.         set.add(n);
4.
5.         for(int i=1;i<n;i++)
6.         {
7.             if(n%i==1)
8.             {
9.                 numberOfDistinctIntegers(n-i);
10.              }
11.          }
12.           return set.size();
13.     }
```

Sample Input:

```
int number=7;
System.out.println("result : "+numberOfDistinctIntegers(number));
```

Sample Output

```
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files
result : 5

Process finished with exit code 0
```