Design and analysis of algorithm ✕ | Design and analysis of algorithm ✕ | +

onlinecourses.nptel.ac.in/noc20_cs27/progassignment?name=100

Apps | Programming | Games | Videos | Assorted | Academics | ToDo | Crackwatch | Flutter | Design and analysis... | Undergraduate Ad... | Computer Organisa...

Week 2 Programming Assignment: Moving Intervals

Due on 2020-02-20, 23:59 IST

- Select your language (C/C++/Java/Python3)
- Paste your code into the submission window.
- There are some public test cases and some (hidden) private test cases.
- "Compile and run" will evaluate your submission against the public test cases.
- "Submit" will evaluate your submission against the hidden private test cases. There are 10 private testcases in all, each with equal weightage. You will get feedback about which private test cases pass or fail, though you cannot see the actual test cases.
- Ignore warnings about "Presentation errors".

Moving Intervals

Zonal Computing Olympiad 2018

There are **C** cakes in a row, numbered from 1 to **C**. There are **N** children, each of whom have selected a consecutive set of cakes to eat. That is, Child i has decided to eat all the cakes from S_i to E_i, end points inclusive. If there is a cake which appears in some two childrens' set, then they will fight because both of them want to eat that cake, and you don't want that to happen.

You will be given an integer **K** which will be either 0 or 1. If **K** is 0, then you should find out if some two children will fight. Print "Good" if no one fights, and "Bad" if someone fights.
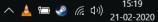
If **K** is 1, then you can persuade at most one child to change his decision to some other set of cakes. But the number of cakes that he eats must be the same. That is, if Child i had initially decided that he wants to eat the cakes from S_i to E_i, then you could persuade the child to instead eat the cakes from X to Y instead, for any valid X and Y (ie. $1 \le X \le Y \le C$), provided that the number of cakes is the same (ie. $E_i - S_i + 1 = Y - X + 1$). If after persuading at most 1 Child to change his decision, no fights happen, then print "Good". But if no matter what you do, someone will fight, then print "Bad".

Solution hint

Sort the intervals and check for overlaps.

Input format

The first line of each test case contains three integers **C**, **N** and **K** denoting the number of cakes, number of children and **K**,

Input format

The first line of each test case contains three integers C, N and K denoting the number of cakes, number of children and K, respectively.

The i-th of the next N lines contains two space separated integers S_i and E_i which denotes the initial decision of Child i. That is, Child i wants to eat from cake S_i to cake E_i.

Output format

For each test case, output a single line containing "Good" or "Bad".

Constraints

- $1 \le T \le 10$
- $1 \le C \le 10^9$
- $1 \le N \le 10^5$
- $0 \le K \le 1$
- $1 \le S_i, E_i \le C$

Sample input 1

```
5 2 0
2 2
3 5
```

Sample output 1

```
Good
```

Sample input 2

```
5 2 1
2 2
```

Design and analysis of algorithm: ×   S Design and analysis of algorithm: ×   +

onlinecourses.nptel.ac.in/noc20_cs27/progassignment?name=100

Apps   Programming   Games   Videos   Assorted   Academics   ToDo   Crackwatch   Flutter   S Design and analysis...   Undergraduate Ad...   Computer Organisa...

Sample input 2

```
5 2 1
2 2
2 5
```

Sample output 2

```
Good
```

Sample input 3

```
5 2 1
2 3
2 5
```

Sample output 3

```
Bad
```

Explanation

Sample 1: Child 1 wants to eat the second cake, and Child 2 wants to eat Cakes 3, 4 and 5. So there is no fight, and the answer is "Good".

Sample 2: Child 1 wants to eat Cake 2, and Child 2 wants to eat Cakes 2, 3, 4 and 5. Both of them want to eat Cake 2, and hence it could lead to a fight. But because **K** = 1, we can persuade one of the children to change their decision. For instance, we could persuade Child 1 to change his decision from [2, 2] to [1, 1]. After this, there is no fight, and the hence answer is "Good".

Sample 3: Child 1 wants to eat Cake 2 and Cake 3, and Child 2 wants to eat Cakes 2, 3, 4 and 5. Both of them want to eat Cakes 2 and 3, and hence it could lead to a fight. And because **K** = 1, we can persuade one of the children to change their decision. For instance, we could persuade Child 1 to change his decision from [2, 3] to [1, 2]. But even after this, both of them want to eat Cake 2. You can verify that no matter how we persuade at most 1 child, they will end up fighting. Hence the answer is "Bad".

S Design and analysis of algorithm ✕ S Design and analysis of algorithm ✕ +

onlinecourses.nptel.ac.in/noc20_cs27/progassignment?name=100

Apps Programming Games Videos Assorted Academics ToDo Crackwatch Flutter S Design and analysis... Undergraduate Ad... Computer Organisa...

Explanation

Sample 1: Child 1 wants to eat the second cake, and Child 2 wants to eat Cakes 3, 4 and 5. So there is no fight, and the answer is "Good".

Sample 2: Child 1 wants to eat Cake 2, and Child 2 wants to eat Cakes 2, 3, 4 and 5. Both of them want to eat Cake 2, and hence it could lead to a fight. But because $K = 1$, we can persuade one of the children to change their decision. For instance, we could persuade Child 1 to change his decision from [2, 2] to [1, 1]. After this, there is no fight, and the hence answer is "Good".
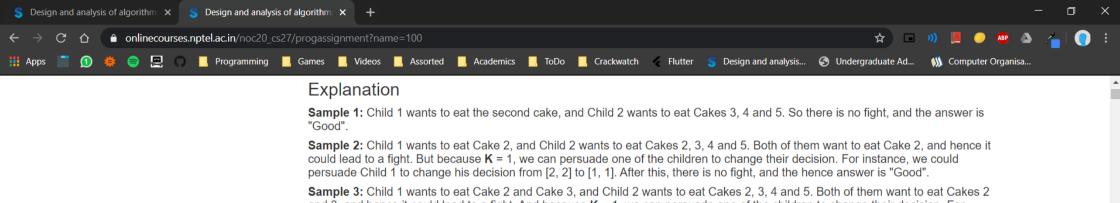
Sample 3: Child 1 wants to eat Cake 2 and Cake 3, and Child 2 wants to eat Cakes 2, 3, 4 and 5. Both of them want to eat Cakes 2 and 3, and hence it could lead to a fight. And because $K = 1$, we can persuade one of the children to change their decision. For instance, we could persuade Child 1 to change his decision from [2, 3] to [1, 2]. But even after this, both of them want to eat Cake 2. You can verify that no matter how we persuade at most 1 child, they will end up fighting. Hence the answer is "Bad".

Private Test cases used for evaluation	Input	Expected Output	Actual Output	Status
	1000000000 1000 0			
	469294915 469339452			
	307159883 307412920			
	384202260 384374190			
	122725181 123064568			
	413461997 413922553			
	448184031 448597285			
	452234955 452704813			
	175278149 175279388			
	112524124 112864015			
	29897537 29949971			
	473921388 474453539			
	66612065 66820444			
	330615556 331019478			
	54084739 54137570			
	130962551 131361424			
	271196329 271522775			
	160579880 160608349			
	134165753 134269793			
	36136035 36917534			
	298324632 298328085			
	315168449 315715245			
	57993064 58016004			
	126815937 126842379			