

X


<https://swayam.gov.in>

https://swayam.gov.in/nc_details/NPTEL

souravsharma2468@gmail.com ▾

NPTEL (<https://swayam.gov.in/explorer?ncCode=NPTEL>) » **Design and analysis of algorithms (course)**

Announcements (announcements)

About the Course (https://swayam.gov.in/nd1_noc20_cs27/preview) Ask a Question (forum)

Progress (student/home) Mentor (student/mentor)

Course outline

How does an NPTEL online course work?

 Week 1 :
Introduction

 Week 1 :
Analysis of algorithms

Week 1 Quiz

 Week 2 :
Searching and sorting

Week 2 Quiz

 Week 2
Programming Assignment

 ● Week 2
Programming Assignment:
Moving

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 (<https://www.iarcs.org.in/inoi/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

Intervals

(/noc20_cs27/progassignment?name=100)

Week 3 : Graphs

Week 3 Quiz

Week 3 Programming Assignment

Week 4 : Weighted graphs

Week 4 Quiz

Week 4 Programming Assignment

Week 5: Data Structures: Union-Find and Heaps

Week 5 : Divide and Conquer

Week 5 Quiz

Week 6: Data Structures: Search Trees

Week 6: Greedy Algorithms

Week 6 Quiz

Week 6 Programming Assignment

Week 7: Dynamic Programming

Week 7 Quiz

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. $1 \leq X \leq Y \leq C$), 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 , 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

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

Sample input 1

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

Sample output 1

```
Good
```

Sample input 2

```
5 2 1
2 2
```

| |
|--|
| Week 7 |
| Programming Assignment |
| Week 8: Linear Programming and Network Flows |
| Week 8: Intractability |
| Week 8 Quiz |
| Text Transcripts |
| Books |
| Download Videos |

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".

| Private Test cases used for evaluation | Input | Expected Output | Actual Output | Status |
|--|--|-----------------|---------------|--------|
| Test Case 1 | 1000000000 1000 0 469294915 4 69339452 307159883 3 07412920 384202260 3 84374190 122725181 1 | Bad\n | Bad\n | Passed |

| | |
|-----------|-----|
| 23064568 | |
| 413461997 | 4 |
| 13922553 | |
| 448184031 | 4 |
| 48597285 | |
| 452234955 | 4 |
| 52704813 | |
| 175278149 | 1 |
| 75279388 | |
| 112524124 | 1 |
| 12864015 | |
| 29897537 | 29 |
| 949971 | |
| 473921388 | 4 |
| 74453539 | |
| 66612065 | 66 |
| 820444 | |
| 330615556 | 3 |
| 31019478 | |
| 54084739 | 54 |
| 137570 | |
| 130962551 | 1 |
| 31361424 | |
| 271196329 | 2 |
| 71522775 | |
| 160579880 | 1 |
| 60608349 | |
| 134165753 | 1 |
| 34269793 | |
| 36136035 | 36 |
| 917534 | |
| 298324632 | 2 |
| 98328085 | |
| 315168449 | 3 |
| 15715245 | |
| 57993064 | 58 |
| 016004 | |
| 126815937 | 1 |
| 26842379 | |
| 309153567 | 3 |
| 09238153 | |
| 380653007 | 3 |
| 81030430 | |
| 188283725 | 1 |
| 88383619 | |
| 137052738 | 1 |
| 37261161 | |
| 143779906 | 1 |
| 44288324 | |
| 5462302 | 553 |
| 0905 | |

| | |
|-----------|------|
| 80178419 | |
| 20091674 | 20 |
| 124961 | |
| 403899443 | 4 |
| 03903926 | |
| 1376570 | 138 |
| 7006 | |
| 396991274 | 3 |
| 97003169 | |
| 171474375 | 1 |
| 71508183 | |
| 345499027 | 3 |
| 45521036 | |
| 272364918 | 2 |
| 72371431 | |
| 227060827 | 2 |
| 27109426 | |
| 242104876 | 2 |
| 42161738 | |
| 273175323 | 2 |
| 73213431 | |
| 329821599 | 3 |
| 29830449 | |
| 230959782 | 2 |
| 30995939 | |
| 79270220 | 79 |
| 296845 | |
| 217768201 | 2 |
| 17774909 | |
| 311863889 | 3 |
| 11884743 | |
| 330080991 | 3 |
| 30093226 | |
| 338726626 | 3 |
| 38789326 | |
| 74168163 | 74 |
| 170581 | |
| 366809073 | 3 |
| 66821330 | |
| 287387180 | 2 |
| 87399995 | |
| 282985 | 2983 |
| 32 | |
| 338958803 | 3 |
| 38991530 | |
| 247442053 | 2 |
| 47457441 | |
| 88030365 | 88 |
| 039333 | |
| 432681079 | 4 |
| 32721296 | |

```

145189026 1
45204747
189269266 1
89330739
335072615 3
35086515
145334079 1
45359267
116142238 1
16191445
473622811 4
73660960
442848516 4
42901604
191846496 1
91872814

```

The due date for submitting this assignment has passed.

9 out of 10 tests passed.

You scored 90.0/100.

Assignment submitted on 2020-02-18, 17:21 IST

Your last recorded submission was :

```

1  #include<bits/stdc++.h>
2  using namespace std;
3
4  #define int long long int
5
6
7  int32_t main(){
8
9      ios::sync_with_stdio(false);
10     cin.tie(0);
11     int c,n,k;
12     cin>>c>>n>>k;
13     vector<pair<int,int> >v;
14     for(int i=0;i<n;++i){
15         int a,b;
16         cin>>a>>b;
17         v.push_back({a,b});
18     }
19
20     sort(v.begin(),v.end());
21
22     if(k==0){
23         int cnt=0;
24         for(int i=0;i<v.size()-1;++i){
25             for(int j=i+1;j<v.size();++j){
26                 if(v[i].second<v[j].first)break;
27                 cnt++;
28             }
29         }
30         if(cnt)cout<<"Bad\n";
31         else cout<<"Good\n";
32     }
33     else{
34         vector<int> gap;
35
36         gap.push_back(v[0].first-2);
37         gap.push_back(c-(v[v.size()-1].second+1));
38
39         for(int i=0;i<v.size()-1;++i){
40             gap.push_back(v[i+1].first-v[i].second-1);
41         }
42     }

```

```
43
44     int cnt=0;
45     vector<int> bad;
46     for(int i=0;i<v.size()-1;++i){
47         for(int j=i+1;j<v.size();++j){
48             if(v[i].second<v[j].first)break;
49             else{
50                 bad.push_back(v[i].second-v[i].first);
51                 cnt++;
52                 break;
53             }
54         }
55     }
56
57
58     if(cnt==0){
59         cout<<"Good\n";
60         return 0;
61     }
62
63     if(bad.size()>1){
64         cout<<"Bad\n";
65     }
66     else{
67         int a=bad[0];
68         for(int i: gap){
69             if(i>=a){
70                 cout<<"Good\n";
71                 return 0;
72             }
73         }
74         cout<<"Bad\n";
75     }
76 }
77 return 0;
78 }
```