

AtCoder

AtCoder Beginner Contest 048: *Upsolving*

Prof. Edson Alves - UnB/FGA

2020

Sumário

1. A - AtCoder *** Contest
2. B - Between a and b...
3. C - Boxes and Candies
4. D - An Ordinary Game

A - AtCoder * Contest**

Problema

Snuke is going to open a contest named “AtCoder s Contest”. Here, s is a string of length 1 or greater, where the first character is an uppercase English letter, and the second and subsequent characters are lowercase English letters.

Snuke has decided to abbreviate the name of the contest as “A x C”. Here, x is the uppercase English letter at the beginning of s .

Given the name of the contest, print the abbreviation of the name.

Constraints

- The length of s is between 1 and 100, inclusive.
- The first character in s is an uppercase English letter.
- The second and subsequent characters in s are lowercase English letters.

Input

Input is given from Standard Input in the following format:

```
AtCoder  $s$  Contest
```

Output

Print the abbreviation of the name of the contest.

Exemplo de entradas e saídas

Entrada

AtCoder Beginner Contest

AtCoder Snuke Contest

AtCoder X Contest

Saída

ABC

ASC

AXC

- Este é um problema bastante simples: basta extrair o primeiro caractere da segunda linha da entrada e inserí-lo na segunda posição da string que corresponde à solução
- O primeiro elemento de uma string pode ser acessado por meio da notação de colchetes (`s[0]`) ou através do método `front()`
- Ambas formas tem complexidade $O(1)$
- Com as garantias da entrada, não é necessário nenhum processamento deste caractere

Solução $O(1)$

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 int main()
6 {
7     string a, b, c;
8     cin >> a >> b >> c;
9
10    string ans { "AxC" };
11    ans[1] = b.front();
12
13    cout << ans << '\n';
14
15    return 0;
16 }
```


B - Between a and b...

Problema

You are given nonnegative integers a and b ($a \leq b$), and a positive integer x . Among the integers between a and b , inclusive, how many are divisible by x ?

Constraints

- $0 \leq a \leq b \leq 10^{18}$
- $1 \leq x \leq 10^{18}$

Input

Input is given from Standard Input in the following format:

a *b* *x*

Output

Print the number of the integers between *a* and *b*, inclusive, that are divisible by *x*.

Exemplo de entradas e saídas

Entrada

Saída

4 8 2

3

0 5 1

6

9 9 2

0

1 10000000000000000000 3

3333333333333333333

- O problema consiste em determinar o número de múltiplos de x no intervalo $[a, b]$
- Seja $m(x, b)$ o número de múltiplos de x em $[1, b]$
- É possível mostrar que

$$m(x, b) = \left\lfloor \frac{b}{x} \right\rfloor$$

- Assim, a solução do problema é $m(x, b) - m(x, a - 1)$
- Cuidado, entretanto, com o caso especial $a = 0$
- Neste caso, a solução é $m(x, b) + 1$, uma vez que zero é múltiplo de qualquer inteiro
- Esta solução tem complexidade $O(1)$

Solução $O(N)$

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4 using ll = long long;
5
6 int main()
7 {
8     ll a, b, x;
9     cin >> a >> b >> x;
10
11     ll p = b / x, q = (a - 1)/x;
12
13     auto ans = a > 0 ? p - q : p + 1;
14
15     cout << ans << '\n';
16
17     return 0;
18 }
```

C – Boxes and Candies

Problema

There are N boxes arranged in a row. Initially, the i -th box from the left contains a_i candies.

Snuke can perform the following operation any number of times:

- Choose a box containing at least one candy, and eat one of the candies in the chosen box.

His objective is as follows:

- Any two neighboring boxes contain at most x candies in total.

Find the minimum number of operations required to achieve the objective.

Constraints

- $2 \leq N \leq 10^5$
- $0 \leq a_i \leq 10^9$
- $0 \leq x \leq 10^9$

Input

Input is given from Standard Input in the following format:

```
 $N$      $x$   
 $a_1$    $a_2$   ...   $a_N$ 
```

Output

Print the minimum number of operations required to achieve the objective.

Exemplo de entradas e saídas

Entrada

Saída

3 3

1

2 2 2

6 1

11

1 6 1 2 0 4

5 9

0

3 1 4 1 5

2 0

10

5 5

Solução

- Como o problema determina que a soma dos doces entre duas caixas adjacentes deve ser, no máximo, igual a x , é sempre ótimo diminuir o máximo possível da caixa que é adjacente a outras duas
- Assim, considere que $s = a_{i-1} + a_i$, com $1 < i \leq N$
- Se $s \leq x$, não é preciso remover nenhum doce
- Se $s > x$, faça $d = s - x$
- Dos d doces a serem removidos, é ótimo remover $k = \min\{d, a_i\}$ doces de a_i
- Isto porque, além de realizar a devida correção, subtrair k doces de a_i pode auxiliar a correção do par de elementos vizinhos a_i e a_{i+1}
- Como todos os pares de elementos devem ser avaliados, esta solução tem complexidade $O(N)$

Solução $O(N)$

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4 using ll = long long;
5
6 ll solve(int N, ll x, vector<ll>& xs)
7 {
8     ll ans = 0;
9
10    for (int i = 1; i < N; ++i)
11    {
12        auto sum = xs[i] + xs[i - 1];
13        auto diff = max(0LL, sum - x);
14
15        ans += diff;
16        xs[i] -= min(diff, xs[i]);
17    }
18
19    return ans;
20 }
21
```

Solução $O(N)$

```
22 int main()
23 {
24     ios::sync_with_stdio(false);
25
26     int N, x;
27     cin >> N >> x;
28
29     vector<ll> xs(N);
30
31     for (int i = 0; i < N; ++i)
32         cin >> xs[i];
33
34     auto ans = solve(N, x, xs);
35
36     cout << ans << '\n';
37
38     return 0;
39 }
```

D - An Ordinary Game

Problema

There is a string s of length 3 or greater. No two neighboring characters in s are equal.

Takahashi and Aoki will play a game against each other. The two players alternately performs the following operation, Takahashi going first:

- Remove one of the characters in s , excluding both ends. However, a character cannot be removed if removal of the character would result in two neighboring equal characters in s .

The player who becomes unable to perform the operation, loses the game. Determine which player will win when the two play optimally.

Constraints

- $3 \leq |s| \leq 10^5$
- s consists of lowercase English letters.
- No two neighboring characters in s are equal.

Input

Input is given from Standard Input in the following format:

```
s
```

Output

If Takahashi will win, print **First**. If Aoki will win, print **Second**.

Exemplo de entradas e saídas

Entrada

aba

abc

abcab

Saída

Second

First

First

- A solução do problema depende do entendimento do desenvolvimento do jogo e dos possíveis cenários que levam à derrota de um jogador
- Primeiramente, observe que não é possível remover os caracteres que ocupam a primeira e a última posição da string
- Assim, há dois cenários possíveis:
 1. os extremos da string são iguais; ou
 2. os extremos da string são distintos
- Um jogador ficará impedido de realizar uma jogada e, consequentemente, perderá o jogo, caso ambos vizinhos de qualquer caractere sejam iguais

- Estes cenários tem duas formas possíveis:

$abab \dots ababa$

ou

$abab \dots abab,$

onde a e b são dois caracteres distintos

- Isto porque, caso a string tenha 3 ou mais caracteres distintos, ainda existirá ao menos um movimento válido
- A quantidade de caracteres destes cenários pode variar, mas não sua paridade (ímpar na primeira forma, par na segunda)

- Se for realizado um número par de movimentos do estado inicial até o estado que leva à derrota, o jogador 2 será o vencedor
- Isto acontece se
 - (i) ambos extremos são iguais e N é ímpar
 - (ii) ambos extremos são diferentes e N é par
- Nos demais cenários, o vencedor será o jogador 1
- Assim, a solução depende apenas do valor de N e dos extremos da string s
- Portanto, a complexidade será $O(1)$

Solução $O(1)$

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 string solve(const string& s)
6 {
7     int x = s.front() == s.back() ? 1 : 0;
8     int y = s.size() % 2;
9
10    return x == y ? "Second" : "First";
11 }
12
13 int main()
14 {
15     string s;
16     cin >> s;
17
18     cout << solve(s) << '\n';
19
20     return 0;
21 }
```

1. AtCoder Beginner Contest 048 – Problem A: AtCoder *** Contest
2. AtCoder Beginner Contest 048 – Problem B: Between a and b...
3. AtCoder Beginner Contest 048 – Problem C: Boxes and Candies
4. AtCoder Beginner Contest 048 – Problem D: An Ordinary Game