Minusy (rozwiązanie)

XIV OIJ, zawody I stopnia, tura ukryta
14 października 2019 – 13 stycznia 2020

Autorzy zadania: **Bartosz Łukasiewicz, Jacek Tomasiewicz**
Opracowanie: **Patryk Czajka, Dawid Jamka**

Olimpiada Informatyczna Juniorów

Wyobraźmy sobie szukane rozwiązanie – ciąg, w którym jest największa możliwa liczba kolejnych plusów. Niektóre z tych plusów powstały przez zamianę par minusów, widać jednak, że cały blok plusów musiał powstać z pewnego spójnego fragmentu znaków.

Szukamy zatem takiego kawałka ciągu wejściowego, w którym da się wszystkie minusy zamienić na plusy, a łączna liczba plusów będzie maksymalna możliwa. Aby to zrobić, podzielmy nasz ciąg na wejściu na *bloki* zawierające te same znaki, np. ciąg $++++++---+++-+-----++--$ dzielimy kolejno na bloki $(+\times 5), (-\times 3), (+\times 3), (-\times 1), (+\times 1), (-\times 4), (+\times 2), (-\times 2)$. Każdy blok z parzystą minusów można zamienić na blok z dwukrotnie mniejszą liczbą plusów (wykonując zamiany kolejno od lewej do prawej). Z kolei z bloku z nieparzystą liczbą minusów zawsze musi pozostać jeden minus, możemy jednak tak pozbyć się reszty, aby ten minus pozostawić na początku albo na końcu bloku.

Zatem żadne rozwiązanie nie może zawierać w całości nieparzystego bloku minusów, może się jednak na nim zaczynać i kończyć. Dokładniej: najlepszy fragment, który da się zamienić na kolejne plusy, może zacząć się albo na początku ciągu, albo od któregoś nieparzystego bloku minusów, z którego zostawimy jeden minus na początku. Z kolei koniec takiego fragmentu to albo koniec całego ciągu, albo ponownie blok z nieparzystą liczbą minusów (tym razem opłaca się zostawić minus na końcu tego bloku). Zatem trzeba sprawdzić wszystkie fragmenty ciągu między każdymi dwoma „nieparzystymi" blokami, dla każdego takiego fragmentu obliczyć liczbę możliwych do uzyskania plusów i wybrać najlepszą z tych możliwości.

Aby to uczynić, przechodzimy pętlą przez kolejne znaki ciągu i utrzymujemy trzy zmienne:

- *aktualny* – długość bloku, w którym właśnie jesteśmy;
- *plusy* – liczbę plusów, jakie zgromadziliśmy od ostatniego nieparzystego bloku, wliczając w to plusy uzyskane z dwóch minusów;
- *najlepsze* – najlepsze rozwiązanie, jakie napotkaliśmy do tej pory;

Jeśli nowy znak jest taki sam, jak poprzedni, wciąż jesteśmy w tym samym bloku, a więc zwiększamy *aktualny* o 1. Jeśli nowy znak jest inny, skończył się blok.

Jeżeli był to blok plusów, dodajemy ich liczbę (*aktualny*) do wartości rozwiązania, które właśnie zbieramy (*plusy*). Jeśli był to blok minusów, dodajemy do rozwiązania połowę ich liczby (*aktualny/2*). Dodatkowo, jeśli był to nieparzysty blok minusów, obecne rozwiązanie kończy się, i porównujemy je z dotychczasowym najlepszym kandydatem (*najlepsze*). Od tego samego nieparzystego bloku zacznie się następne możliwe rozwiązanie – czyli wartość *plusy* ustawia się na *aktualny/2*.

Koniec całego ciągu traktujemy jak koniec bloku – dodajemy wartość aktualnego bloku do rozwiązania i porównujemy z najlepszym kandydatem.

min.py

```
def main():
  # Wczytujemy ciąg dany na wejściu.
  ciag = input()
  # Wyliczamy długość tego ciągu.
  n = len(ciag)
  # Deklarujemy pomocnicze zmienne.
  aktualny, plusy, najlepsze = 1, 0, 0

  # Iterujemy się po ciągu,
  for i in range(n):
    # Jeżeli jesteśmy w tym samym bloku, zwiekszamy wartość aktualny.
    if i+1 < n and ciag[i] == ciag[i+1]:
      aktualny += 1
    # W przeciwnym wypadku zakończyliśmy blok.
    else:
      # Jeżeli to był blok plusów,
      if ciag[i] == '+':
        # dodajemy liczbę plusów do rozwiązania.
```


Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

```
        plusy += aktualny
      # W przeciwnym wypadku (blok minusów),
      else:
        # dodajemy do rozwiązania połowę ich liczby.
        plusy += aktualny // 2
        # Dodatkowo jeżeli to był nieparzysty ciąg minusów,
        if aktualny % 2 == 1:
          # Być może znaleźliśmy maksymalne rozwiązanie.
          najlepsze = max(najlepsze, plusy)
          # Po czym mówimy, że od tego bloku może się zacząć kolejne rozwiązanie.
          plusy = aktualny // 2
      # Jako że blok się skończył, ustawiamy liczbę elementów w bloku na 1.
      aktualny = 1
  # Dodatkowo obsługujemy koniec ciągu w podobny sposób jak wyżej.
  plusy += aktualny // 2
  najlepsze = max(najlepsze, plusy)

  # Finalnie wypisujemy wynik.
  print(najlepsze)


main()
```

min.cpp

```
#include "bits/stdc++.h"

using namespace std;

int main() {
  // Wczytujemy ciąg dany na wejściu.
  string ciag;
  cin >> ciag;
  // Wyliczamy długość tego ciągu.
  int n = ciag.size();

  // Deklarujemy pomocnicze zmienne.
  int aktualny = 1, plusy = 0, najlepsze = 0;

  // Iterujemy się po ciągu,
  for (int i=0; i<n; i++) {
    // Jeżeli jesteśmy w tym samym bloku, zwiekszamy wartość aktualny.
    if (i+1 < n and ciag[i] == ciag[i+1]) aktualny += 1;
    // W przeciwnym wypadku zakończyliśmy blok.
    else {
      // Jeżeli to był blok plusów,
      if (ciag[i] == '+') {
        // dodajemy liczbę plusów do rozwiązania.
        plusy += aktualny;
      }
      // W przeciwnym wypadku (blok minusów),
      else {
        // dodajemy do rozwiązania połowę ich liczby.
        plusy += aktualny / 2;
        // Dodatkowo jeżeli to był nieparzysty ciąg minusów,
        if (aktualny % 2 == 1) {
          // Być może znaleźliśmy maksymalne rozwiązanie.
          najlepsze = max(najlepsze, plusy);
          // Po czym mówimy, że od tego bloku może się zacząć kolejne rozwiązanie.
          plusy = aktualny / 2;
```


Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

```
36          }
37        }
38        // Jako że blok się skończył, ustawiamy liczbę elementów w bloku na 1.
39        aktualny = 1;
40      }
41    }
42    // Dodatkowo obsługujemy koniec ciągu w podobny sposób jak wyżej.
43    plusy += aktualny / 2;
44    najlepsze = max(najlepsze, plusy);
45
46    // Finalnie wypisujemy wynik.
47    cout << najlepsze << "\n";
48  }
```


Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".