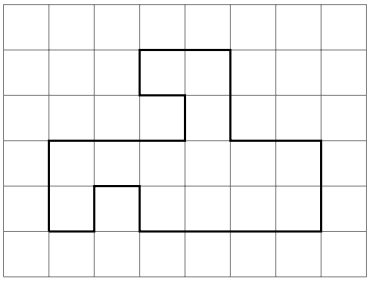
Pole figury II (rozwiązanie)

Autor zadania: **Karol Pokorski**
Opracowanie: **Dominik Klemba, Karol Farbiś, Dominik Wawszczak**
Opis rozwiązania: **Bartosz Kostka**

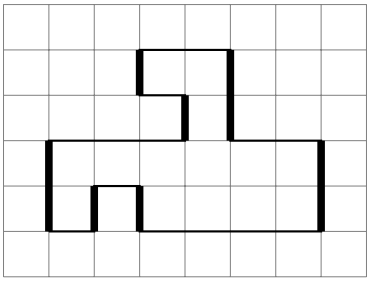
Olimpiada Informatyczna Juniorów

Rozpocznijmy od zaznaczenia, że zadanie to można rozwiązać na wiele sposobów. My jednak skupimy się na rozwiązaniu, które naszym zdaniem jest najprostsze do zaimplementowania.

Rozważmy figurę z przykładu przedstawionego w treści zadania:

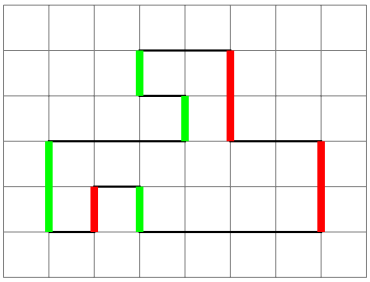


Figurę tę będziemy analizować wiersz po wierszu, skupiając się w każdym wierszu jedynie na odcinkach pionowych (idących w górę lub w dół). Na poniższym rysunku zostały one pogrubione.



W każdym wierszu, analizując odcinki pionowe od lewej do prawej, oznaczymy je jako *otwierające* i *zamykające*. Odcinek otwierający oznacza, że dalej (bezpośrednio po jego prawej stronie) znajduje się wnętrze figury. Natomiast odcinek zamykający oznacza, że bezpośrednio za nim znajdujemy się poza figurą.

Pierwszy odcinek z lewej zawsze będzie odcinkiem otwierającym, natomiast drugi odcinek będzie odcinkiem zamykającym (ponieważ do momentu jego napotkania znajdowaliśmy się wewnątrz figury). Kolejne odcinki (jeśli istnieją) będą pojawiały się naprzemiennie jako otwierające i zamykające. Na poniższym rysunku odcinki otwierające zaznaczyliśmy kolorem zielonym, a zamykające – czerwonym.



Aby obliczyć wkład danego wiersza do pola całej figury, wystarczy wygenerować odcinki pionowe w danym wierszu, posortować je i policzyć odległości między odpowiadającymi sobie odcinkami otwierającymi i zamykającymi. Na przykład ostatni (dolny) wiersz na rysunku powyżej dodaje do pola figury 5 jednostek, ponieważ odległość między pierwszą parą odcinków wynosi 1 (od pierwszego zielonego do drugiego czerwonego), a odległość między drugą parą odcinków wynosi 4 (od trzeciego zielonego do czwartego czerwonego). Zatem $1 + 4 = 5$.

Sumując wkłady ze wszystkich wierszy, otrzymujemy pole całej figury.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".

pol.cpp

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

using namespace std;

int main() {
  // Wczytujemy opis.
  string opis;
  cin >> opis;

  // Początkowe współrzedne ustalamy na (0,0).
  int x = 0, y = 0;

  // I trzymamy mapę odcinków pionowych, gdzie kluczem jest współrzędna y,
  // a wartościami współrzędne x (konkretniej oznacza ona odcinek od (x,y) do
  // (x,y+1)).
  map<int, vector<int>> odcinki_pionowe;
  for (char litera : opis) {
    if (litera == 'G') {
      // Dla ruchu w górę, dodajemy odcinek (x,y) do (x,y+1).
      odcinki_pionowe[y].push_back(x);
      // Po czym przesuwamy współrzędną y o 1 w górę.
      y++;
    } else if (litera == 'D') {
      // Dla ruchu w górę, dodajemy odcinek (x,y-1) do (x,y).
      odcinki_pionowe[y - 1].push_back(x);
      // Po czym przesywamy współrzędną y o 1 w dół.
      y--;
    } else if (litera == 'P') {
      // Dla odcinków poziomych nie zapisujemy ich, tylko
      // odpowiednio modyfikujemy współrzędną x.
      x++;
    } else if (litera == 'L') {
      x--;
    }
  }

  // Finalnie liczymy pole figury.
  // Zwróć uwagę, że pole może być większe niż zakres typu int,
  // zatem używamy zmiennej typu long long.
  long long pole = 0;
  // Iterujemy się po wszystkich współrzędnych y, które zapisaliśmy.
  for (auto &[y, odcinki] : odcinki_pionowe) {
    // Sortujemy listę odcinków.
    sort(odcinki.begin(), odcinki.end());
    // Dodajemy odległość pomiędzy odcinkiem otwierającym (odcinki[i])
    // i zamykającym figurę (odcinki[i+1]).
    for (int i = 0; i < odcinki.size(); i += 2) {
      pole += odcinki[i + 1] - odcinki[i];
    }
  }

  // Finalnie wypisujemy wynik.
  cout << pole << "\n";
}
```


Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".

pol.py

```python
from collections import defaultdict

def main():
    # Wczytujemy opis.
    opis = input()

    # Początkowe współrzędne ustalamy na (0,0).
    x, y = 0, 0

    # Trzymamy mapę odcinków pionowych, gdzie kluczem jest współrzędna y,
    # a wartościami są współrzędne x (konkretniej oznacza ona odcinek od (x,y) do
    # (x,y+1)).  Struktura defaultdict automatycznie inicjalizuje listy.
    odcinki_pionowe = defaultdict(list)

    for litera in opis:
        if litera == "G":
            # Dla ruchu w górę, dodajemy odcinek (x, y) do (x, y+1).
            odcinki_pionowe[y].append(x)
            # Przesuwamy współrzędną y o 1 w górę.
            y += 1
        elif litera == "D":
            # Dla ruchu w dół, dodajemy odcinek (x, y-1) do (x, y).
            odcinki_pionowe[y - 1].append(x)
            # Przesuwamy współrzędną y o 1 w dół.
            y -= 1
        elif litera == "P":
            # Dla ruchu w prawo, modyfikujemy współrzędną x.
            x += 1
        elif litera == "L":
            # Dla ruchu w lewo, modyfikujemy współrzędną x.
            x -= 1

    # Finalnie liczymy pole figury.
    pole = 0

    # Iterujemy po wszystkich współrzędnych y, które zapisaliśmy.
    for y, odcinki in odcinki_pionowe.items():
        # Sortujemy listę odcinków.
        odcinki.sort()
        # Dodajemy odległość pomiędzy odcinkiem otwierającym (odcinki[i])
        # i zamykającym figurę (odcinki[i+1]).
        for i in range(0, len(odcinki), 2):
            pole += odcinki[i + 1] - odcinki[i]

    # Finalnie wypisujemy wynik.
    print(pole)


if __name__ == "__main__":
    main()
```


Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2022/2023, 2023/2024, 2024/2025".