

---

# PPPD - Lab. 05

Copyright ©2021 M. Śleszyńska-Nowak i in.

Zadanie punktowane, lab 05, 2019/2020

---

## Temat: Sad Bożydara

### Treść zadania

Rolnik Bożydar ma prostokaatny sad, w którym drzewka rosną w niektórych punktach o współrzędnych całkowitych. Niestety nikczemnicy kradną mu jabłka, dlatego z troską Bożydar postanowił ogrodzić swoje drzewka płotem. Konieczne jest, żeby płot był ładny - ładny płot jest prostokątem o bokach wzdłuż osi układu współrzędnych. Bożydar chciałby wiedzieć, jak długi będzie ten płot i musimy mu w tym pomóc.

Masz daną funkcję:

```
def drzewo(x, y):
    if x >= 0 and x <= 10 and y >= 0 and y <= 10:
        return (x == y) and (x % 10 < 4 or x % 10 > 5)
    elif x >= -10 and x <= -1 and y >= 0 and y <= 10:
        return True
    else:
        return False
```

która odpowiada na pytanie, czy w punkcie o współrzędnych  $(x, y)$  znajduje się drzewo.

1. Napisz funkcję `wypisz_prostokat(x_min, x_max, y_min, y_max)`, która wypisze podany prostokąt w formacie `[xmin,xmax]x[ymin,ymax]` (Na przykład: `[4,8]x[5,10]`).
2. Wczytaj od użytkownika informacje o położeniu sadu Bożydara: minimalną i maksymalną wartość współrzędnych  $x$  i  $y$ . Wypisz ją używając funkcji z poprzedniego podpunktu.
3. Napisz funkcję `najmniejszy_plot(x_min, x_max, y_min, y_max)` która jako argument przyjmie prostokąt opisujący wymiary sadu Bożydara i zwróci krotkę opisującą wymiary najkrótszego płotu zawierającego wszystkie drzewa (przypomnijmy: krotkę konstruujemy poprzez napis `(a,b,c,d)`). W szczególnym przypadku, kiedy wewnątrz podanego prostokąta nie ma żadnego drzewka lub podano niepoprawny protokąt, zwróć prostokąt `[0,0]x[0,0]`. Następnie wywołaj tę funkcję dla podanych przez użytkownika danych i wypisz wynik używając funkcji `wypisz_prostokat`.

Przykład: dla sadu `[0,5]x[2,4]` z następującym układem drzew:

```
4   .D....
3   ..DD..
2   ...D..
y/x 012345
```

(gdzie D oznacza drzewo, współrzędna  $x$  rośnie w prawo i jest w zakresie od 0 do 5, a współrzędna  $y$  rośnie w górę i jest w zakresie od 2 do 4) rozwiązaniem jest prostokąt `[1,3]x[2,4]`. Wskazówka: wystarczy wyznaczyć najmniejszą i największą wartość każdej współrzędnej.

Wskazówka dotycząca krotek:

```
def dzielenie(a, b):
```

```

return a//b, a%b

def main():
    # odbiór wyniku do dwóch zmiennych,
    # tzw. rozpakowywanie krotki
    iloraz, reszta = dzielenie(5,3)

```

**UWAGA:** rysunek jest poglądowy i nie musi przedstawiać rozkładu drzew zwracanego przez funkcję `drzewo()`.

4. Napisz funkcję `zapisz_do_pliku(sciezka, x_min, x_max, y_min, y_max)`, która w pliku wskazanym w parametrze `sciezka` zapisuje powyższy rysunek (czyli rozkład drzew na pełnym sadzie Bożydara wraz z podpisanymi osiami), wraz ze współrzędnymi. Oczywiście musimy powtórzyć bardzo podobne obliczenia jak w funkcji w poprzednim podpunkcie. Osie powinny wyglądać ładnie i się zgrywać z rysunkiem dla jednocyfrowych liczb nieujemnych.
5. Napisz funkcję `obwod_prostokata(x_min, x_max, y_min, y_max)`, która wyznaczy obwód podanego prostokąta. Używając tej funkcji wyznacz i wypisz obwód sadu Bożydara oraz długość znalezionej płotu.
6. Bożydar postanowił podzielić swój sad na dwa mniejsze w nadziei, że uda się je ogrodzić krótszym płotem. Sad zostanie podzielony na dwa prostokąty po współrzędnej  $x$  - tzn. dla pewnego  $x^*$  będą to prostokąty  $[x_{min}, x^*] \times [y_{min}, y_{max}]$  oraz  $[x^*+1, x_{max}] \times [y_{min}, y_{max}]$ . Napisz funkcję `najlepszy_podzial(x_min, x_max, y_min, y_max)` która przyjmie prostokąt opisujący wymiary sadu i zwróci wartość  $x^*$  wybraną w taki sposób, żeby suma długości płotów potrzebnych do ogrodzenia obu części sadu była jak najmniejsza. Wywołaj tę funkcję w mainie i wypisz znaną wartość  $x^*$ .

Wskazówka: wystarczy dla każdej poprawnej wartości  $x^*$  sprawdzić sumę obwodów obu części, używając wcześniej napisanych funkcji `najmniejszy_plot` oraz `obwod_prostokata`.

Przykład: W powyższym przykładzie  $x^*$  może mieć wartości 0, 1, 2, 3, 4 i 5. Jeżeli weźmiemy  $x^*=1$ , to pierwsza część sadu będzie zawierała jedno drzewo, które ogrodzimy prostokątem  $[1,1] \times [4,4]$  (o obwodzie 0), a druga część sadu zostanie ogrodzona prostokątem  $[2,3] \times [2,3]$  (o obwodzie 4). Można sprawdzić, że każdy inny podział da większą sumę obwodów, więc odpowiedzią jest 1.

7. Napisz własną funkcję `drzewo2(x, y)`, która działa w sposób następujący: jeśli współrzędne wychodzą poza prostokąt  $[-10, 10] \times [-10, 10]$ , zwraca `False`. Jeśli jesteśmy w prostokącie  $[0, 10] \times [-10, 10]$ , to zwracamy `True` z prawdopodobieństwem 0.5. Jeśli jesteśmy w prostokącie  $[-10, -1] \times [-10, 10]$ , to zwracamy `True` z prawdopodobieństwem 0.1.

Uwaga: Program powinien być napisany w taki sposób, aby działał poprawnie po zmianie funkcji `drzewo()` na dowolną inną (tzn. jedyny dopuszczalny sposób sprawdzania, czy w jakimś punkcie znajduje się drzewo, to wywołanie funkcji `drzewo()`).

Przykłady interakcji użytkownika z programem:

```

Podaj wymiary sadu w kolejnosci xmin, xmax, ymin, ymax:
0
10
0
10
Podany prostokąt to:
[0, 10]x[0, 10]
Najmniejszy płot zawierający wszystkie drzewka:
[0, 10]x[0, 10]
Obwód sadu: 40
Obwód płotu: 40
Najlepszy podział to 3

```

---

## Punktacja

Za poszczególne etapy można uzyskać następującą liczbę punktów:

- pkt 1,2,3 - 2p
- pkt 4 - 2p
- pkt 5. i 6. - 4p
- pkt 7 - 2p

### Uwaga

- Jeśli program się nie kompiluje (interpretuje), ocena jest zmniejszana o połowę.
- Jeśli kod programu jest niskiej jakości (nieestetycznie formatowanie, mylące nazwy zmiennych itp.), ocena jest zmniejszana o 2 p.