**Języki Skryptowe**

Dokumentacja projektu PIONEK

Kacper Grabiec, grupa 2C

Wydział Matematyki Stosowanej

Informatyka

**Część I**

Obraz zawierający stół

Opis wygenerowany automatycznie**Opis programu**

**Instrukcja obsługi**

W celu rozwiązania zadania należy uruchomić plik *bat.bat*. Po jego włączeniu ukaże się nam menu z kilkoma opcjami do wyboru.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Program czeka, aż wybierzemy jedną z dostępnych opcji:

* **Stwórz raport** – program rozwiązuje problem dla zadanych danych wejściowych oraz tworzy raport na podstawie danych wyjściowych;
* **Wyświetl ostatni raport** – program wyświetla nam najnowszy plik *.html* zawierający gotowy raport;
* **Usuń raporty** – program usuwa wszystkie raporty znajdujące się w folderze */raporty*;
* **Uruchom skrypt py1.py** – program uruchamia tylko plik *py1.py*, odpowiedzialny za rozwiązanie zadania;
* **Uruchom skrypt py2.py** – program uruchamia tylko plik *py2.py*, odpowiedzialny za stworzenie raportu;
* **Koniec** – koniec działania programu;

Po wykonaniu każdej opcji menu wyświetla się ponownie.

Każdy plik zawierający dane wejściowe (np. *in1.txt*) znajduje się w folderze */input\_data*. Jego zawartość wygląda następująco:

Obraz zawierający tekst

Opis wygenerowany automatycznie

Pierwszy wiersz standardowego wejścia zawiera jedną dodatnią liczbę całkowitą *n* oznaczającą liczbę możliwych ruchów pionka. Każdy z kolejnych *n* wierszy zawiera dwie liczby całkowite *x*, *y* oddzielone pojedynczym odstępem i oznaczające wektor [*x, y*] opisujący możliwy ruch pionka.

Każdy plik zawierający dane wyjściowe (n. *out1.txt*) znajduje się w folderze */output\_data*. Jego zawartość wygląda następująco:



Program wypisuje na standardowe wyjście liczbę całkowitą oznaczającą kwadrat odległości od punktu (0*,* 0) do najdalszego punktu, do którego może doskoczyć pionek.

Wygenerowany raport wygląda następująco:

Obraz zawierający stół

Opis wygenerowany automatycznie

Po lewej stronie wyświetlone zostają kolejne pliki zawierające dane wejściowe, po prawej dane wyjściowe.

**Część II**

**Opis działania**

Zadanie polega na wyznaczeniu liczby całkowitej będącą kwadratem odległości od punktu *(0, 0)* do najdalszego punktu, do którego może doskoczyć pionek. Wartość tą uzyskamy ze wzoru:

Jako, że potrzebujemy kwadrat odległości, wynik otrzymamy w następujący sposób:

Ostateczny wektor zmienia się w zależności od tego ile ruchów wykorzystamy. W tym celu musimy także dodawać następny wektory ruchu. Można to zrobić według następującego wzoru:

**Algorytm**

Obraz zawierający tekst

Opis wygenerowany automatycznie

**Implementacja**

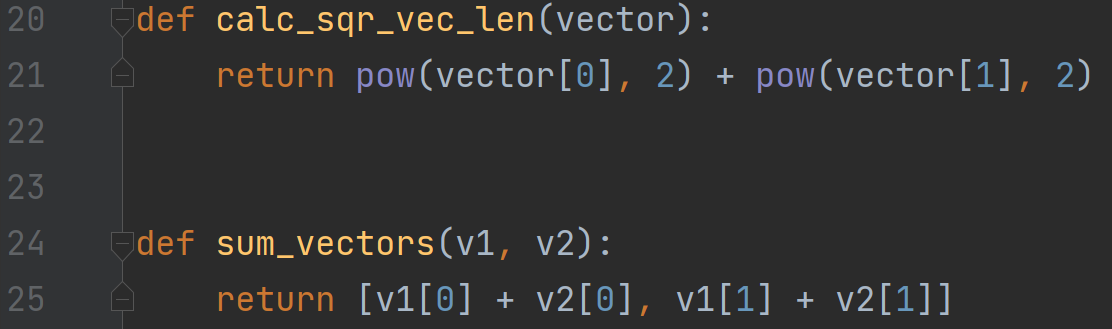
**Plik *py1.py***

Obraz zawierający tekst

Opis wygenerowany automatycznie

Powyższe funkcje służą do odczytywania danych z pliku oraz do ich zapisywania. Funkcja *load\_data()* otwiera plik i zwraca zawartość w postaci tablicy.

Funkcja *print\_data()* zapisuje otrzymane wyniki do pliku.



Funkcje matematyczne służące do obliczenia długości wektora oraz do dodawania wektorów.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Funkcja *solve()* jest najważniejszą funkcją w programie. Jest ona wywoływana rekurencyjnie. Sprawdza, czy dany ruch nie został już wykorzystany a następnie oblicza dystans. Rekurencja przerywa się gdy wszystkie ruchy zostaną wykorzystane.

**Plik *py2.py***

Obraz zawierający tekst

Opis wygenerowany automatycznie

Funkcja *load\_data()* wczytuje zawartości wszystkich plików w danym katalogu i zwraca je w postaci tablicy.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Funkcja *generate\_template()* dla każdych danych wejściowych oraz wyjściowych tworzy osobny element *HTML*.

Obraz zawierający tekst

Opis wygenerowany automatycznie

Funkcja *print\_data()* wczytuje obecną datę oraz czas. Następnie tworzy podstawowy szablon *HTML*, który zestawia dane wejściowe z danymi wyjściowymi.

**Pełen kod aplikacji**

<https://github.com/KacperGrabiec/JezykiSkryptoweProjekt>