

Języki skryptowe

dokumentacja projektu Zlicz Literki

Bartosz Jarzyński, grupa 3E

2 grudnia 2021

Część I

Opis programu

XIX OI, Etap I - Zadanie litery

Mały Jaś ma bardzo długie nazwisko. Nie jest jednak jedyną taką osobą w swoim środowisku. Okazało się bowiem, że jedna z jego koleżanek z przedszkola, Małgosia, ma nazwisko dokładnie tej samej długości, chociaż inne. Co więcej, ich nazwiska zawierają dokładnie tyle samo liter każdego rodzaju - tyle samo liter A, tyle samo liter B, itd.

Jaś i Małgosia bardzo się polubili i często bawią się razem. Jedną z ich ulubionych zabaw jest zebranie dużej liczby małych karteczek, napisanie na nich kolejnych liter nazwiska Jasia, a następnie przesuwanie karteczek tak, aby powstało z nich nazwisko Małgosi.

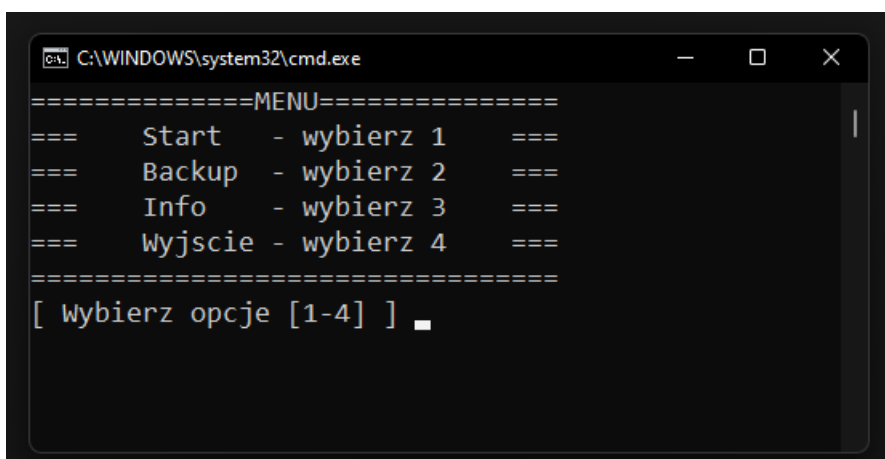
Ponieważ Jaś uwielbia łamigłówki, zaczął zastanawiać się, ile co najmniej zamian sąsiednich liter trzeba wykonać, żeby przekształcić jego nazwisko w nazwisko Małgosi. Nie jest to łatwe zadanie dla kilkuletniego dziecka, dlatego Jaś poprosił Ciebie, głównego programistę w przedszkolu, o napisanie programu, który znajdzie odpowiedź na nurtujące go pytanie.

W pierwszym wierszu standardowego wejścia znajduje się jedna liczba całkowita n ($2 \leq n \leq 1000000$) oznaczająca liczbę liter w nazwisku Jasia. W drugim wierszu znajduje się n kolejnych liter nazwiska Jasia (bez odstępów). W trzecim wierszu znajduje się n kolejnych liter nazwiska Małgosi (również bez odstępów). Oba napisy składają się jedynie z wielkich liter alfabetu angielskiego.

Twój program powinien wypisać na standardowe wyjście jedną liczbę całkowitą, oznaczającą minimalną liczbę zamian sąsiednich liter, które przekształcają nazwisko Jasia w nazwisko Małgosi.

Instrukcja obsługi

Program uruchamiamy plikiem o nazwie skrypt.bat. Możemy tenże plik wywołać również z poziomu wiersza poleceń. Jest to plik wykonywalny, w którym operujemy programem całościowo. Po włączeniu programu wita nas poniższy komunikat.



```
C:\WINDOWS\system32\cmd.exe

====MENU====
=== Start - wybierz 1 ===
=== Backup - wybierz 2 ===
=== Info - wybierz 3 ===
=== Wyjscie - wybierz 4 ===
[ Wybierz opcje [1-4] ] _
```

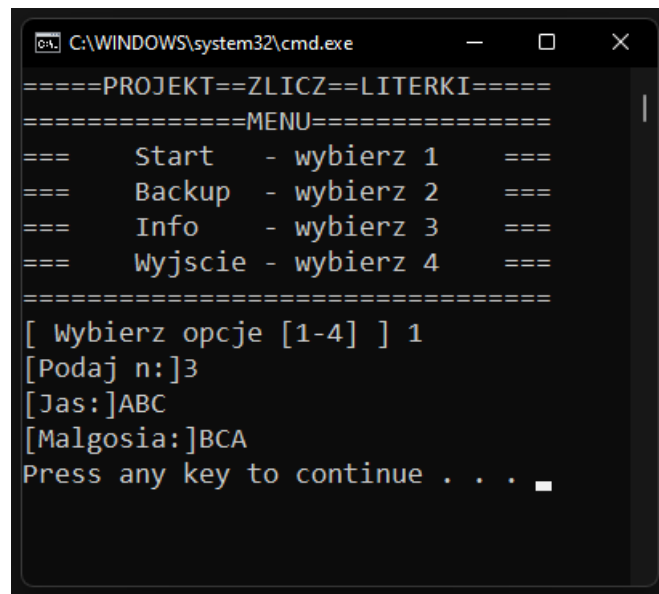
Rysunek 1: Ekran początkowy skryptu programu.

1. Start

Przy wybraniu opcji pierwszej, czyli START, program uruchomi się. W jego działanie zalicza się kolejno:

- Zadeklarowanie przez użytkownika wartości *n*
- Zadeklarowanie przez użytkownika nazwiska Jasia
- Zadeklarowanie przez użytkownika nazwiska Małgosi

Następnie skrypt przetworzy wpisane przez użytkownika dane i zapisze je do utworzonego przez skrypt pliku tekstowego o nazwie *input.txt*. Z tego właśnie pliku pobierane są dane wejściowe do programu. Następnie uruchamiany jest *main.py*, czyli plik wykonywalny, pobiera on dane i kompiluje program. W tym samym czasie skrypt uruchamia stronę internetową WWW, na której podane są informacje na temat autora, tematu zadania oraz dane wejściowe, wyjściowe oraz przykładowa statystyka - czas, w którym program wykonał wszystkie obliczenia, podane w sekundach.



```
C:\WINDOWS\system32\cmd.exe

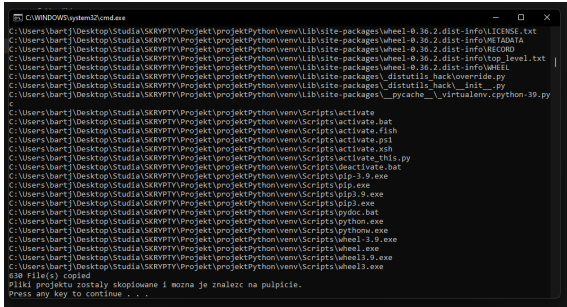
=====PROJEKT==ZLICZ==LITERKI=====
=====MENU=====
===      Start   - wybierz 1      ===
===      Backup  - wybierz 2      ===
===      Info    - wybierz 3      ===
===      Wyjscie - wybierz 4      ===
=====

[ wybierz opcje [1-4] ] 1
[Podaj n:]3
[Jas:]ABC
[Malgosia:]BCA
Press any key to continue . . .
```

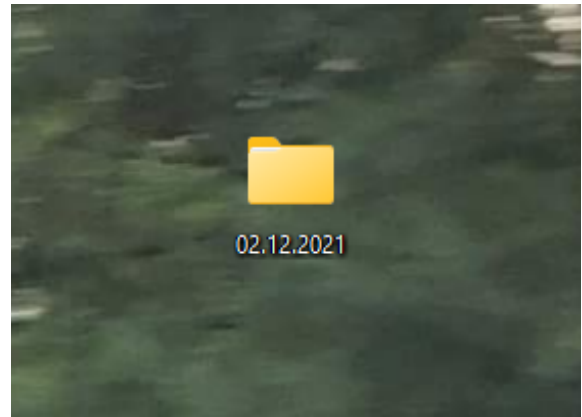
Rysunek 2: Wykonanie startujące program. Podane przeze mnie dane to przykładowe dane, sprawdzające z treści zadania.

2. Backup

Przy wybraniu opcji drugiej, czyli BACKUP, program wykona kopię zapasową plików umieszczonych w katalogu projektu. Kopia będzie znajdować się w nowo utworzonym katalogu na pulpicie, którego nazwa jest równoznaczna z obecną datą, na przykład 2-12-2021 - jak na poniższym przykładzie:

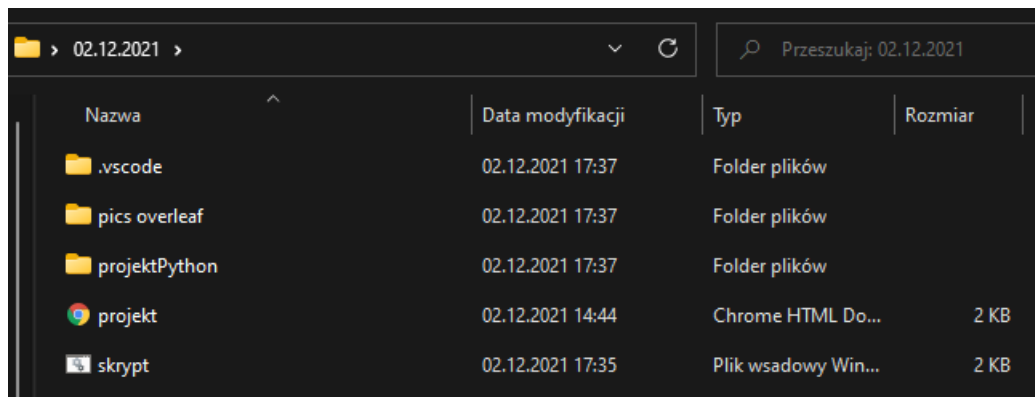


(a) Wykonanie kopii zapasowej.



(b) Utworzony folder *Backup* z dzisiejszą datą.

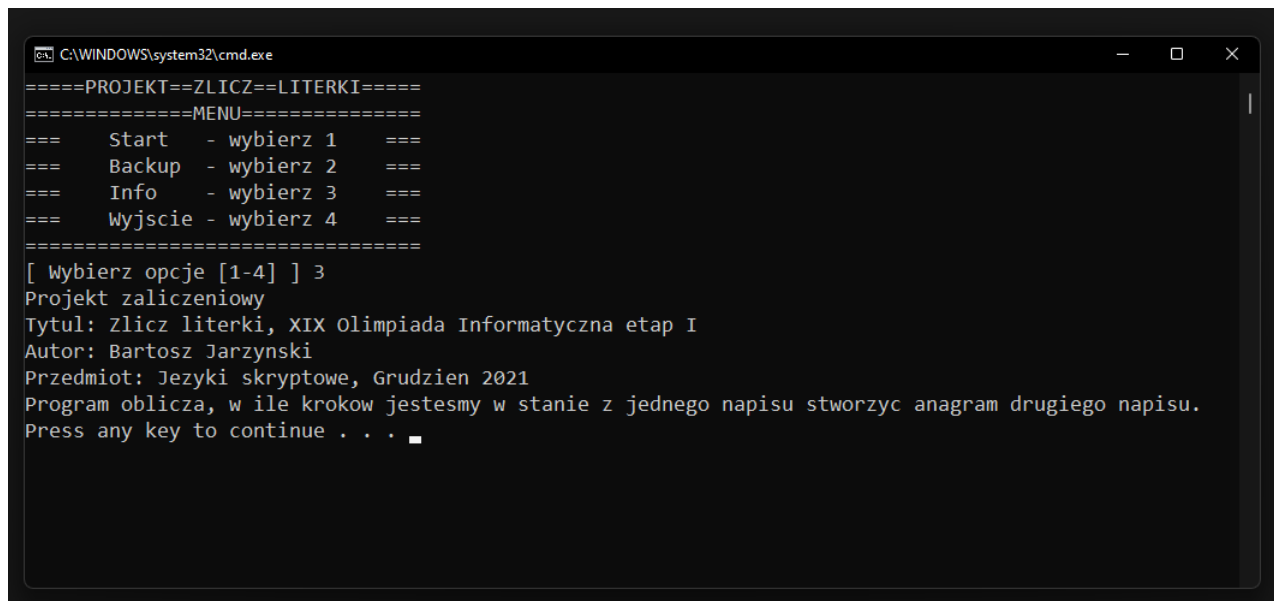
Rysunek 3: Działanie funkcji Backup'u.



Rysunek 4: Skopiowane pliki w folderze *Backup*.

3. Informacje

Przy wybraniu opcji trzeciej, czyli INFO, program pokazuje informacje o projekcie i jego krótkim opisie. Nie zabrakło informacji o autorze, tytule oraz samym projekcie.



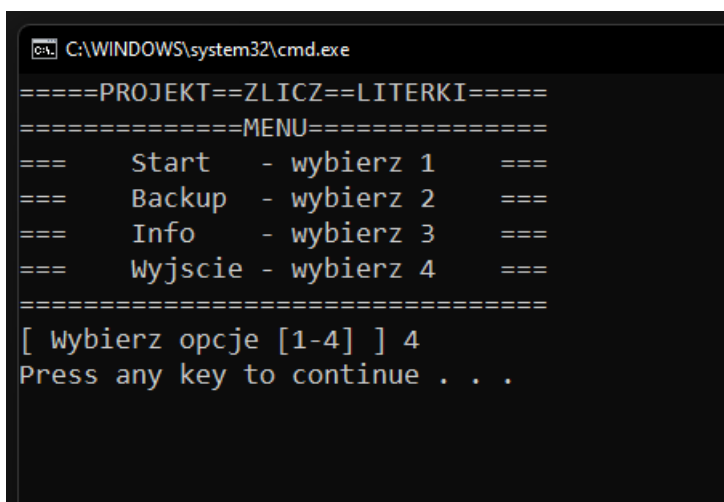
```
C:\WINDOWS\system32\cmd.exe

=====PROJEKT==ZLICZ==LITERKI=====
=====MENU=====
===      Start   - wybierz 1      ===
===      Backup  - wybierz 2      ===
===      Info    - wybierz 3      ===
===      Wyjscie - wybierz 4      ===
=====
[ Wybierz opcje [1-4] ] 3
Projekt zaliczeniowy
Tytuł: Zlicz literki, XIX Olimpiada Informatyczna etap I
Autor: Bartosz Jarzynski
Przedmiot: Języki skryptowe, Grudzien 2021
Program oblicza, w ile krokow jestesmy w stanie z jednego napisu stworzyc anagram drugiego napisu.
Press any key to continue . . .
```

Rysunek 5: Informacje o projekcie w skrypcie Batch.

4. Wyjście

Ostatnia dostępna opcja to *Wyjście*. Jej użycie spowoduje wyłączenie wiersza poleceń, tym samym programem.



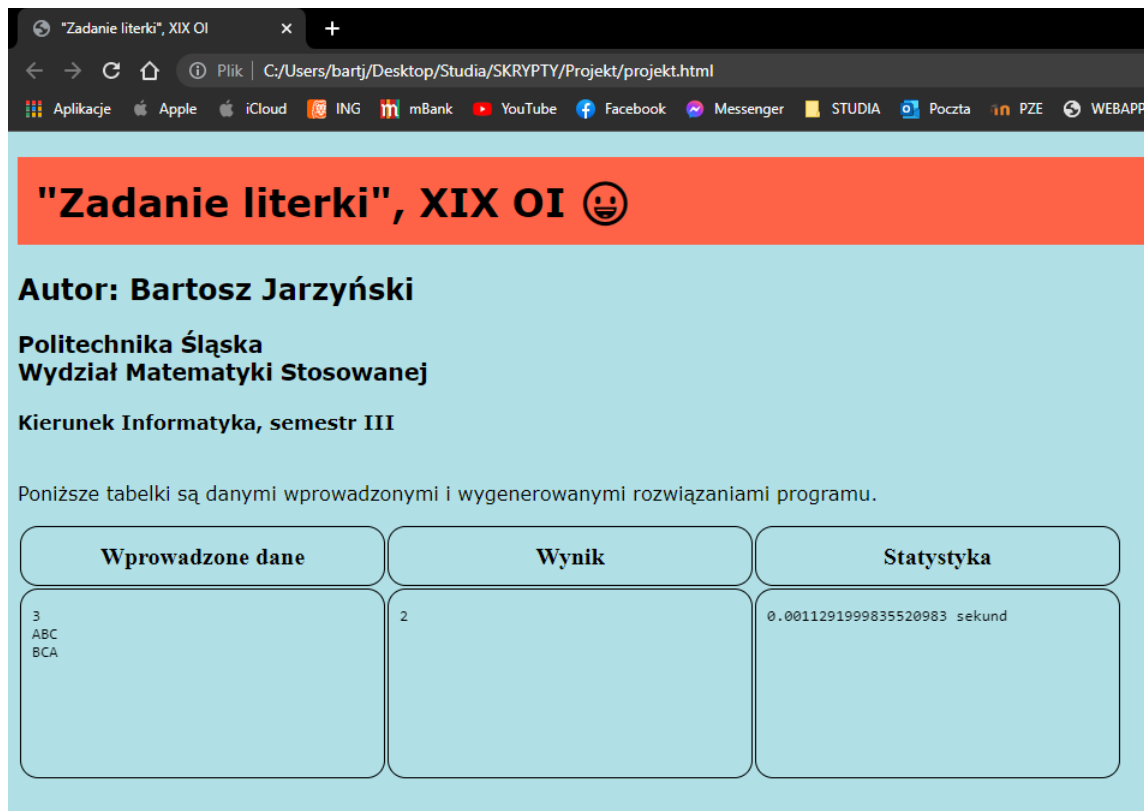
```
C:\WINDOWS\system32\cmd.exe

=====PROJEKT==ZLICZ==LITERKI=====
=====MENU=====
===      Start   - wybierz 1      ===
===      Backup  - wybierz 2      ===
===      Info    - wybierz 3      ===
===      Wyjscie - wybierz 4      ===
=====
[ Wybierz opcje [1-4] ] 4
Press any key to continue . . .
```

Rysunek 6: Wyjście z programu.

5. Rezultaty kompilacji

Na nowo powstałej stronie internetowej WWW zobaczymy dane wejściowe, wynik oraz statystykę programu, czyli czas do wykonania obliczeń przez komputer. Dodatkowo strona utrzymana jest w przyjaznym designie oraz zawarte zostały tam informacje o zadaniu i autorze.



Rysunek 7: Strona internetowa stworzona na potrzeby zadania.

Dodatkowe informacje

Dla poprawnego działania stworzonego programu projektowego potrzebne jest zainstalowane środowisko języka Python. Zalecana jest wersja 3 lub wyższa, jeśli jest dostępna.

Część II

Opis działania

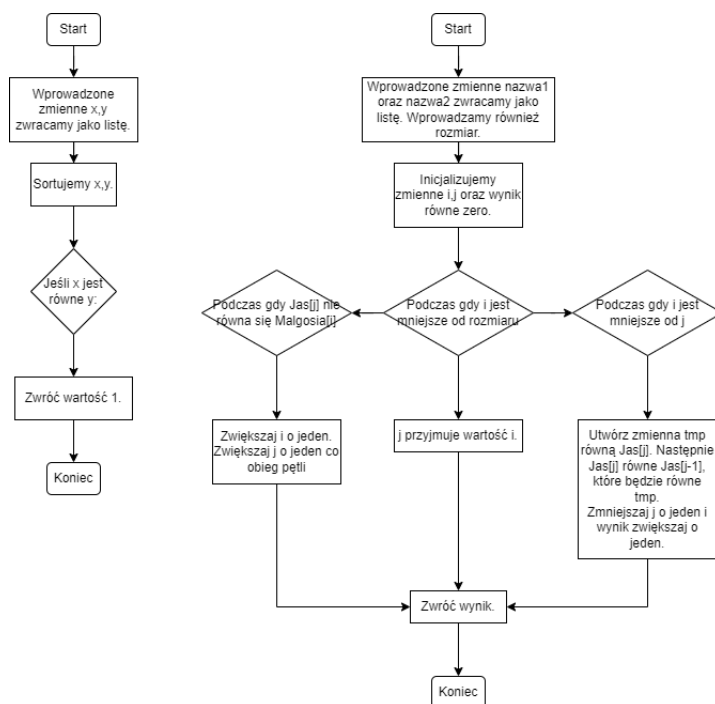
Algorytm pobiera od użytkownika następujące dane:

- Długość obu nazwisk, nazw - oznaczane w programie jako n ,
- Nazwisko Jasia, nazwę pierwszą - oznaczane w programie jako Jas ,
- Nazwisko Małgosi, nazwę drugą - oznaczane w programie jako $Malgosia$.

Te trzy dane posłużą nam do wykonania działania algorytmu oraz do uzyskania odpowiedzi na pytanie z treści zadania.

Program w swoim działaniu na początku sprawdza, czy przypadkiem podane przez nas nazwy nie są sobie równe. W tym celu tworzy dwie listy, posortowane alfabetycznie. Jeśli tak się stanie, program zwróci wartość 1 lub 0, w zależności, czy musiał przeprowadzić jakąkolwiek operację. Jeżeli obie nazwy są anagramami i ich wprowadzona długość zgadza się, to znaczy nie jest inna niż długość nazwiska Jasia lub Małgosi - program zaczyna kompilowanie.

Program tworzy dwie listy nazw, inicjalizuje potrzebne do algorytmu zmienne i rozpoczyna działanie. Zastosowany został algorytm sortowania, który przestawia litery tak długo, aż nie znajdzie ciągu podanych liczb pasującego do drugiego nazwiska. Na koniec zwraca wynik, czyli liczbę sygnalizującą, ile zmian program musiał wykonać. Na sam koniec program liczy statystykę, czyli ile komputer potrzebował do wygenerowania wyniku. Wszystkie dane są pobierane oraz wypisywane do plików tekstowych, zapisywanych w folderze projektu.



Rysunek 8: Schemat blokowy przedstawionego powyżej algorytmu.

Algorytm

Pseudokod tworzymy w \LaTeX . Przykład:

Definicja 1:

Data: Dane wejściowe Stringi $txt1$ oraz $txt2$

$x := sort(txt1);$

$y := sort(txt2);$

if $x == y$ **then**

 | return 1;

end

Result: return 0;

Definicja 2:

Data: Dane wejściowe String $Nazwa1$, $Nazwa2$, int $Rozmiar$

$jas := list(Nazwa1);$

$malgosia := list(Nazwa2);$

$i, j, wynik := 0;$

while $i < Rozmiar$ **do**

 | $j := i;$

while $jas[j] \neq malgosia[i]$ **do**

 | $j++$

end

while $i < j$ **do**

 | $tmp := jas[j];$

 | $jas[j] := jas[j - 1];$

 | $jas[j - 1] := tmp;$

 | $j--$

 | $wynik++$

end

 | $i++$

end

Result: return wynik;

Algorithm 1: Algorytm szukania najszybszej drogi do utworzenia anagramu.

Implementacja pliku wejściowego

Aby program mógł zadziałać potrzebujemy danych wejściowych. Dane podawane są z wysokości wiersza poleceń przez użytkownika, podczas wybrania opcji *Start*. Skrypt tworzy plik *input.txt*, a program usuwa wszelkie znaki białe oraz odczytuje linia po linii kolejno rozmiar, nazwisko pierwsze i nazwisko drugiego. Nie ma możliwości podania więcej niż te trzy dane. Następnie sprawdzamy instrukcję warunkową z zadania, czy n znajduje się w poprawnym przedziale oraz czy jest takiej samej wartości co długość nazwiska Jasia lub Małgosi. Jeśli zdarzy się, że n jest różne od długości któregoś z nazwisk, program wypisze do pliku *output.txt* błąd.

```
1 plikinput = open("input.txt","r")
2 text = plikinput.read()
3
4 lines = text.split("\n")
5 lines = [line.replace(' ', '') for line in lines]
6
7 n = int(lines[0])
8 jasiek = str(lines[1])
9 malgorzata = str(lines[2])
10
11 if n < 1 or n > 1000001 or n!=len(jasiek) or n!=len(malgorzata):
12     error = open("output.txt", "w")
13     error.write("Bład podania parametru n lub n!=jasiek/malgosia")
14     error.close()
```

Pliki wejścia:

Data: Dane wejściowe *input.txt*

Otwieramy i czytamy kolejne linie z pliku;

Rozdzielamy tekst spacjami oraz usuwamy wszystkie białe znaki;

Ustawiamy zmienne n , *Jasiek* oraz *Malgorzata*;

if $n < 1$ or $n > 1000001$ or $n \neq \text{len}(\text{Jasiek})$ or $n \neq \text{len}(\text{Malgorzata})$;

then

 | Wydrukuj Error.;

end

Algorithm 2: Pseudokod dot. pliku wejścia.

Implementacja pliku wyjściowego

Jeśli wszystkie dane zgadzają się, wtedy wykonywana jest funkcja kalkująca kroki. Następnie wynik jest wypisywany do nowo utworzonego pliku *output.txt*. Jeśli program napotka jakikolwiek błąd w pliku zobaczymy wydrukowaną wartość -1.

```
1 if (zadanieLiterki(jasiek, malgorzata)):
2    plik = open("output.txt", "w")
3    plik.write(str(kroki(jasiek, malgorzata, n)))
4    plik.close()
5 else:
6     print(-1)
```

Pliki wejścia:

Data: Dane wejściowe z *input.txt*

if *Funkcja działa i dane są poprawne:* **then**

 Otwórz plik *output.txt*;

Result: Nadpisz w nim rezultat obliczeń

 ;

end

Wydrukuj -1;

Algorithm 3: Pseudokod dot. pliku wyjścia.

Implementacja pliku statystycznego

Za każdym razem, bez znaczenia czy program skompiluje się poprawnie czy nie, komputer obliczy, ile zajęło mu czasu, aby wyprowadzić wynik. Do tego potrzebna była zaimplementowana biblioteka *time* oraz ustalenia zmiennej *czas*.

```
1 import time
2
3 start_time = time.perf_counter()
4 czas = time.perf_counter() - start_time
5
6 stat = open("statystyka.txt", "w")
7 stat.write(str(czas) + " sekund")
8 stat.close()
```

Pliki wejścia:

Data: Biblioteka *time*;

Ustal czas początkowy jako czas pracy programu;

Ustal czas trwania pracy programu jako różnicę zakończenia pracy do rozpoczęcia pracy programu;

Otwórz plik *statystyka.txt*

Result: Nadpisz w nim rezultat obliczeń

Algorithm 4: Pseudokod dot. pliku ze statystykami.

Testy

Jako wzorzec do uzyskania poprawnych wyników przyjąłem przykładowe dane z zadania, to jest:

- Rozmiar $n = 3$,
- Nazwisko Jasia jako ABC,
- Nazwisko Małgosi jako BCA.

Według wzorca odpowiedzią powinna być liczba 2, ponieważ dokonujemy przestawienia litery 'A' o dwa miejsca w celu uzyskania nazwiska pierwszego.

| "Zadanie literki", XIX OI 😊 | | |
|---|-------|------------------------------|
| Autor: Bartosz Jarzyński | | |
| Politechnika Śląska | | |
| Wydział Matematyki Stosowanej | | |
| Kierunek Informatyka, semestr III | | |
| Poniższe tabelki są danymi wprowadzonymi i wygenerowanymi rozwiązaniami programu. | | |
| Wprowadzone dane | Wynik | Statystyka |
| 3 ABC BCA | 2 | 0.0006017000414431095 sekund |

Rysunek 9: Dane wytworzone w programie, usytuowane na stronie WWW.

Wynik został wygenerowany poprawnie, co dowodzi, iż zadanie zostało wykonane pomyślnie oraz prawidłowo.

Pełen kod Python

```
1 import time
2 start_time = time.perf_counter()
3
4 def zadanieLiterki(txt1, txt2):
5
6     x = list(txt1)
7     y = list(txt2)
8
9     x = x.sort()
10    y = y.sort()
11
12    if (x == y):
13        return 1
14    return 0
15
16 def kroki(nazwa1, nazwa2, rozmiar):
17     jas = list(nazwa1)
18     malgosia = list(nazwa2)
19
20     i = 0
21     j = 0
22     wynik = 0
23
24     while (i < rozmiar):
25         j = i
26
27         while (jas[j] != malgosia[i]):
28             j += 1
29
30         while (i < j):
31             tmp = jas[j]
32             jas[j] = jas[j - 1]
33             jas[j - 1] = tmp
34             j -= 1
35             wynik += 1
36         i += 1
37     return wynik
38
39 plikinput = open("input.txt", "r")
40 text = plikinput.read()
41
42 lines = text.split("\n")
43 lines = [line.replace(' ', '') for line in lines]
44
45 n = int(lines[0])
46 jasiak = str(lines[1])
47 malgorzata = str(lines[2])
48
49 if n < 1 or n > 1000001 or n != len(jasiak) or n != len(malgorzata):
50
51     error = open("output.txt", "w")
52     error.write("Bład podania parametru n lub n!=jasiak/malgosia")
```

```
53     error.close()
54
55 elif (zadanieLiterki(jasiek, malgorzata)):
56
57     plik = open("output.txt", "w")
58     plik.write(str(kroki(jasiek, malgorzata, n)))
59     plik.close()
60
61 else:
62     print(-1)
63
64 czas = time.perf_counter() - start_time
65
66 stat = open("statystyka.txt", "w")
67 stat.write(str(czas) + " sekund")
68 stat.close()
```

Pełen kod HTML

```
1 <!DOCTYPE html>
2
3 <html>
4   <head>
5     <style>
6       body {background-color:powderblue;}
7       .important{ background-color: tomato;
8         color:black;
9         padding: 15px;}
10      h1,h2,h3,h4,p {font-family:verdana}
11      th,td {border:1px solid black;
12        border-radius: 15px;
13        text-align: center;}
14
15    </style>
16
17    <title>"Zadanie literki", XIX OI</title>
18
19  </head>
20
21  <body>
22
23    <h1 class="important">"Zadanie literki", XIX OI &#128512;</h1>
24    <h2>Autor: Bartosz Jarzynski</h2>
25    <h3>Politechnika Slaska<br>Wydzial Matematyki Stosowanej</h3>
26    <h4>Kierunek Informatyka, semestr III</h4>
27
28    <p><br>Ponizsze tabelki sa danymi wprowadzonymi i
29      wygenerowanymi rozwiazaniami programu.</p>
30
31    <table style="width:700px">
32
33      <tr style="height:50px;">
34        <th style="font-size: 20px;">Wprowadzone dane</th>
35        <th style="font-size: 20px;">Wynik</th>
36        <th style="font-size: 20px;">Statystyka</th>
37      </tr>
38
39      <tr>
40        <td><div><object data="C:\Users\bartj\Desktop\Studia\
41          SKRYPTY\Projekt\projektPython\input.txt"></object>
42          </div></td>
43        <td><div><object data="C:\Users\bartj\Desktop\Studia\
44          SKRYPTY\Projekt\projektPython\output.txt"></object>
45          </div></td>
46        <td><div><object data="C:\Users\bartj\Desktop\Studia\
47          SKRYPTY\Projekt\projektPython\statystyka.txt"></object>
48          </div></td>
49      </tr>
```

```
44         </table>
45
46     </body>
47 </html>
```

Pełen kod BATCH

```
1 @echo off
2
3 :menu
4 cls
5
6 echo =====PROJEKT==ZLICZ==LITERKI=====
7 echo =====MENU=====
8 echo ===      Start      - wybierz 1      ===
9 echo ===      Backup    - wybierz 2      ===
10 echo ===      Info      - wybierz 3      ===
11 echo ===      Wyjscie    - wybierz 4      ===
12 echo =====
13
14 set /p variable=[ Wybierz opcje [1-4] ]
15 IF %variable%==1 GOTO startt
16 IF %variable%==2 GOTO backup
17 IF %variable%==3 goto info
18 IF %variable%==4 goto exitt
19
20 :startt
21 set /P n=[Podaj n:]
22 set /P Jas=[Jas:]
23 set /P Malgosia=[Malgosia:]
24 cd C:\Users\bartj\Desktop\Studia\SKRYPTY\Projekt\projektPython
25 echo %n% > input.txt
26 echo %Jas% >> input.txt
27 echo %Malgosia% >> input.txt
28 cd C:\Users\bartj\Desktop\Studia\SKRYPTY\Projekt\projektPython
29 start main.py
30 start chrome.exe C:\Users\bartj\Desktop\Studia\SKRYPTY\Projekt\projekt.
    html
31
32 pause
33 cd C:\Users\bartj\Desktop\Studia\SKRYPTY\Projekt\projektPython
34 del input.txt
35 del statystyka.txt
36 del output.txt
37
38 goto menu
39 :backup
40 cd C:\Users\bartj\Desktop
41 SET DT=%date:/=-%
42 mkdir %DT%
43 xcopy C:\Users\bartj\Desktop\Studia\SKRYPTY\Projekt C:\Users\bartj\
    Desktop\%DT% /E /H /C /I
44 echo Pliki projektu zostaly skopiowane i mozna je znalezc na pulpicie.
45 pause
46
47 goto menu
48 :info
49 echo Projekt zaliczeniowy
50 echo Tytul: Zlicz literki, XIX Olimpiada Informatyczna etap I
```

```
51 echo Autor: Bartosz Jarzynski
52 echo Przedmiot: Jezyki skryptowe, Grudzien 2021
53 echo Program oblicza, w ile krokow jestesmy w stanie z jednego napisu
    stworzyc anagram drugiego napisu.
54 pause
55
56 goto menu
57 :exitt
58 pause
```
