

# Języki Skryptowe

dokumentacja projektu Liczby zaprzyjaźnione

Marcin Ryt, grupa 4/8

22 grudnia 2022

# Część I

## Opis programu

Za Wikipedią: "Liczby zaprzyjaźnione to para różnych liczb naturalnych, takich, że suma dzielników każdej z tych liczb równa się drugiej (nie uwzględniając tych dwóch liczb jako dzielników)." Np. liczba 284 ma dzielniki: 1, 2, 4, 71, 142, których suma daje 220, a liczba 220 ma dzielniki: 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110, których suma daje 284. Zatem liczby 220 i 284 tworzą parę liczb zaprzyjaźnionych. Należy napisać program, który dla dowolnej pary różnych liczb naturalnych będzie rozstrzygał, czy para ta tworzy liczby zaprzyjaźnione.

## Instrukcja obsługi

Aby uruchomić program należy włączyć skrypt Zaprzyjaznione.bat otwierający menu obsługi programu. Po uruchomieniu wyświetli nam się instrukcja obsługi programu.

```
-----Menu-----
1. Uruchom program
2. Wyświetl informacje
3. Backup
4. Zakończ
-----
Wybierz numer polecenia (1,2,3,4) _
```

Rysunek 1: Główne menu programu

Możliwe są następujące wybory:

1. Uruchom program - Uruchamia program który pobiera dane z pliku input, sprawdza czy są zaprzyjaźnione, zapisuje je w pliku dane.txt, następnie zwraca komunikat, po czym tworzy raport.html.

```
Wybierz numer polecenia (1,2,3,4) 1
Podaj pierwszą liczbę naturalną: 1
Podaj drugą liczbę naturalną: 2
('Liczby ', 1, ' i ', 2, ' nie sa zaprzyjaznione')
Press any key to continue . . . _
```

Rysunek 2: Przykładowy komunikat po wpisaniu liczb przez użytkownika

## Raport z godziny: 21-12-2022 23:53:37

Input	Output
1	('Liczby ', 1, ' i ', 2, ' nie sa zaprzyjaznione')
2	

Marcin Ryt

Rysunek 3: Przykładowy raport programu

W innym przypadku wypisze komunikat o ponownym wpisaniu liczb

```
Wybierz numer polecenia (1,2,3,4) 1
Podaj pierwszą liczbę naturalną: abc
Liczba naturalna to liczba całkowita większa niż 0
Podaj pierwszą liczbę naturalną: 0
Liczba naturalna to liczba całkowita większa niż 0
Podaj pierwszą liczbę naturalną: -2
Liczba naturalna to liczba całkowita większa niż 0
Podaj pierwszą liczbę naturalną: 2.2
Liczba naturalna to liczba całkowita większa niż 0
Podaj pierwszą liczbę naturalną:
```

Rysunek 4: Przykładowy komunikat o błędnym podaniu liczby

2. Wyświetl informacje - Wypisuje na ekran konsoli opis programu

```
Wybierz numer polecenia (1,2,3,4) 2
Program dla zadanych liczb naturalnych sprawdza czy te liczby sa "zaprzyjaznione".
Liczby zaprzyjaznione to para liczb ktorych suma dzielnikow pierwszej liczby rowna sie drugiej liczbie,
a suma dzielnikow drugiej liczby rowna sie pierwszej.

Program zwraca komunikat w zaleznosci czy liczby sa zaprzyjaznione czy tez nie.

Autor projektu: Marcin Ryt grupa 4/8
```

Rysunek 5: Informacje o programie

3. Backup - Tworzy kopię zapasową danych w katalogu Backup który zawiera raport.html jak i plik dane.txt z inputem i outputem

```
C:dane.txt
1 File(s) copied
```

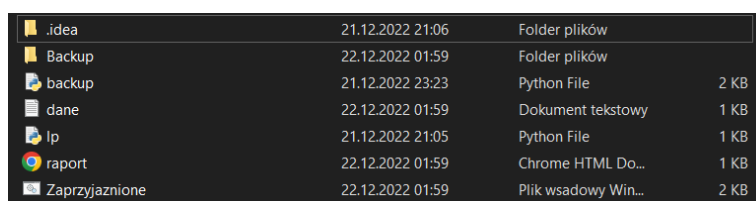
Rysunek 6: Informacja o zkopowania plików do folderu Backup

4. Zakoncz - Zamyka menu, kończy program.

## Struktura danych programu

Do prawidłowego uruchomienia aplikacji program wymaga następujących struktur danych:

- Zaprzyjzaznione.bat - Skrypt batch uruchamiający menu, dzięki niemu możemy uruchomić funkcję wyświetlającą informację o programie, utworzyć backup plików i uruchomić polecenie skryptów projektowych
- lp.py - Skrypt python pobierający od użytkownika 2 liczby naturalne i sprawdzający czy są one liczbami zaprzyjzaznionymi, powoduje zapis do pliku dane.txt.
- raport.py - Skrypt python pobierający dane z pliku dane.txt, generuje raport bazujący na inpucie i outpucie
- dane.txt - Plik tekstowy zawierający input, output jak i komunikat związany z wynikiem programu.
- Katalog .idea zawierający enkodowania plików
- Katalog Backup zawierający pliki dane.txt i raport.html



.idea	21.12.2022 21:06	Folder plików	
Backup	22.12.2022 01:59	Folder plików	
backup	21.12.2022 23:23	Python File	2 KB
dane	22.12.2022 01:59	Dokument tekstowy	1 KB
lp	21.12.2022 21:05	Python File	1 KB
raport	22.12.2022 01:59	Chrome HTML Do...	1 KB
Zaprzyjzaznione	22.12.2022 01:59	Plik wsadowy Win...	2 KB

Rysunek 7: Zrzut ekranu plików znajdujących się w folderze projektu

## Część II

### Opis działania

Skrypt `Zaprzyjajzone.bat` pobiera od użytkownika cyfry odpowiadającą jednej z czterech funkcji wybranych przez użytkownika. Pierwsza funkcja uruchamia program `lp.py` który pobiera z pliku `input` liczby z warunkiem, że są one naturalne po czym liczy sumy dzielników tych liczb i sprawdza czy suma dzielników liczby pierwszej równa się drugiej liczbie i czy suma dzielników drugiej liczby równa się drugiej. Następnie program zwraca komunikat czy podane przez użytkownika liczby są zaprzyjaźnione czy też nie. Liczby jak i komunikat zostają zapisane do pliku `dane.txt` z którego `backup.py` pobiera dane i tworzy z nich `raport.html` który zawiera tabelę z zawartością `output` oraz `input`. Na końcu uruchamiana jest przeglądarka użytkownika w której wyświetla się `raport.html`

### Algorytm

**Data:** Dane wejściowe liczba  $k$   
**Result:** Brak  
 $asuma := 0;$   
 $bsuma := 0;$   
 $i := 0;$   
 $j := 0;$   
**for**  $i < a$  **do**  
    **if**  $a \% i == 0$  **then**  
         $asuma += i$   
    **end**  
**end**  
**for**  $j < b$  **do**  
    **if**  $b \% j == 0$  **then**  
         $bsuma += j$   
    **end**  
**end**  
**if**  $a == bsuma$  **and**  $b == asuma$  **then**  
    Wydrukuj informację, że liczby  $a$  i  $b$  są zaprzyjaźnione;  
**else**  
    Wydrukuj informację, że liczby  $a$  i  $b$  nie są zaprzyjaźnione;  
**end**  
Przypisz  $a, b, output$  do pliku `dane.txt`

**Algorithm 1:** Algorytm sprawdzania czy para liczb jest zaprzyjaźniona.

## Implementacja systemu

Skrypt lp.py pobiera od użytkownika dwie liczby i sprawdza czy są naturalne i zaprzyjaźnione po czym wypisuje komunikat.

Opcja druga wyświetlająca informacje o projekcie

W skrypcie Zaprzyjaźnione.bat po kliknięciu funkcji Backup, skrypt usuwa folder Backup po czym tworzy go na nowo kopiując do niego pliki: dane.txt, raport.html. Każde ponowne wpisanie danych do programu zmienia zawartość pliku dane.txt a co za tym idzie wewnątrz raport.html. Dodatkowo uruchomienie plik backup.py tworzy za każdym nowym inputem tabelę o specjalnej stylistyce.

### 0.1 Wykorzystane biblioteki z opisem

```
1      //Biblioteka pozwalająca na usuwanie plików
2      import os
3      os.remove("raport.html")
4      //Usuniecie pliku raport.html
```

---

```
1      //Biblioteka pozwalająca na uzyskanie terażniejszej daty
2      from datetime import datetime
3      now=datetime.now()
4      fulldate = now.strftime("%d-%m-%Y %H:%M:%S")
5      //Przypisanie do fulldate Daty aktualnej w sekwencji dzien-
        miesiac-rok godzina-minuta-sekunda
```

---

```
1      //Biblioteka pozwalająca na sprawdzenie czy dany plik istnieje
2      from os.path import exists
3      if exists("raport.html"):
4          print("True")
5      //Jezeli plik raport.html istnieje to wypisze komunikat "True"
```

---

## 0.2 Funkcje zawarte w lp.py

```
1 //Funkcja zwracajaca sume dzielnikow podanej liczby
2 def sum_dziel(n):
3     suma = 0
4     for i in range(1, (n // 2 + 2)):
5         if n % i == 0:
6             suma += i
7     return suma
```

---

```
1
2 // Funkcja sprawdzajaca czy liczba jest naturalna
3 def podaj_liczbe(ktora):
4     while 1 > 0:
5         try: // Proba wpisania liczby zgodn z wartosciami integer
6             a = int(input("Podaj " + ktora + " liczb naturaln : "))
7             if a > 0:
8                 return a
9             else:
10                raise ValueError //Wywo anie warunku except
11        except:
12            print("Liczba naturalna to liczba calkowita wieksza ni 0")
```

---

## 0.3 Funkcje zawarte w backup.py

```
1 //Funkcja otwierajaca i zapisujaca plik do listy dla podanej sciezki
2 def otw(file):
3     fil = open(file, "r")
4     for i in fil:
5         l1.append(i)
6     fil.close()
```

---

```
1 //Funkcja tworzaca raport.html
2 def whatever(file):
3     if exists("raport.html"): //Sprawdzanie czy istnieje plik raport.
4         html
5         os.remove("raport.html") //Usuniecie pliku raport.html
6     now=datetime.now()
7     fulldate = now.strftime("%d-%m-%Y %H:%M:%S") //Pobranie aktualnego
8     czasu wedlug dnia-miesiaca-roku godziny-minuty-sekundy
9     otw(file)
10    fil2=open("raport.html","w")//otworzenie pliku i przygotowanie do
11    zapisu
12 //Utworzenie stylu tabeli i wczytanie poszczegolnych danych
13 fil2.write("""
14 <!DOCTYPE html>
15 <html>
16     <head>
17         <style>
18             table, th, td {
19                 border: 1px solid black;
20             }
```

```

18         </style>
19     </head>
20     <body>
21
22         <h1>Raport z godziny: ""+fulldate+""</h1>
23
24         <table>
25             <tr>
26                 <th>Input</th>
27                 <th>Output</th>
28             </tr>
29             <tr>
30                 <td>""+str(l1[0]) +""</td>
31                 <td rowspan="2">""+str(l1[2]) +""</td>
32             </tr>
33             <tr>
34                 <td>""+str(l1[1]) +""</td>
35             </tr>
36         </table>
37         <h4>Marcin Ryt</h4>
38     </body>
39 </html>
40 """)
41 fil2.close()//Zamkniecie pliku

```

---

## Testy

Dane wczytane:

a=220

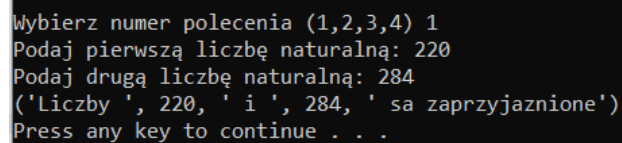
b=284

Dane wyjściowe:

220

284

('Liczby ', 220, ' i ', 284, ' sa zaprzyjaznione')



```

Wybierz numer polecenia (1,2,3,4) 1
Podaj pierwszą liczbę naturalną: 220
Podaj drugą liczbę naturalną: 284
('Liczby ', 220, ' i ', 284, ' sa zaprzyjaznione')
Press any key to continue . . .

```

Rysunek 8: Test działania programu w konsoli



## Raport z godziny: 22-12-2022 03:51:08

Input	Output
220	('Liczby ', 220, ' i ', 284, ' sa zaprzyjaznione')
284	

Marcin Ryt

Rysunek 9: Raport

### Eksperymenty

Podczas realizacji zadania zacząłem szukać pewnych optymalizacji kodu dotyczących wyszukiwania liczb zaprzyjaznionych w podanym zakresie, niestety nie udało mi się znaleźć najbardziej zoptymalizowanego algorytmu który wykorzystuje mało pamięci RAM

```
1 def sum_dziel(n):
2     suma=0
3     for i in range(1,(n//2+2)):
4         if n%i==0:
5             suma+=i
6     return suma
7 l1=[]
8 for i in range(1000,1300):
9     for j in range(i//2,2*i):
10        if sum_dziel(j)==i and sum_dziel(i)==j and i!=j:
11            l1.append(sorted([i,j]))
12 l2=[]
13 for i in l1:
14     if i not in l2:
15         l2.append(i)
16 print(l2)
```

---

# Pełen kod aplikacji

## lp.py

Program lp.py pobiera liczby z pliku input.txt po czym sprawdza czy zgadzają się one z kryteriami liczb naturalnych, jeżeli coś się nie zgadza to wyrzuca błąd, natomiast jeżeli wszystko jest dobrze to zapisuje liczby do listy l1. Następnie pobiera liczby z listy l1 i używając funkcji sum\_dziel(n) liczy sumy dzielników poszczególnych liczb i sprawdza czy są one zaprzyjaźnione. W kolejnym kroku otwiera plik tekstowy dane i zapisuje dane wraz z komunikatem output.

```
1 def sum_dziel(n):
2     suma = 0
3     for i in range(1, (n // 2 + 2)):
4         if n % i == 0:
5             suma += i
6     return suma
7 l1=[]
8 def podaj_liczbe(sciezka):
9     f=open(sciezka,"r")
10    for i in f:
11        try:
12            if int(i)>0:
13                l1.append(int(i))
14            else:
15                raise ValueError
16        except:
17            print("Poda e z e dane")
18            break
19    f.close()
20 podaj_liczbe("input.txt")
21 print(l1)
22
23 filename="dane.txt"
24 f2=open(filename, "w")
25 for j in range(1, (len(l1)/2)+2,2):
26     if l1[j-1]==sum_dziel(l1[j]) and l1[j]== sum_dziel(l1[j-1]) and l1[j-1]!=l1[j]:
27         output="Liczby ", l1[j-1], " i ", l1[j], " sa zaprzyjaznione"
28     else:
29         output="Liczby ", l1[j-1], " i ", l1[j], " nie sa zaprzyjaznione"
30
31     f2.write(str(l1[j-1]))
32     f2.write('\n')
33     f2.write(str(l1[j]))
34     f2.write('\n')
35     f2.write(str(output))
36     f2.write('\n')
37     print(output)
38 f2.close()
```

---

## backup.py

Program backup.py na początku sprawdza czy istnieje plik raport.html, jeżeli istnieje to go usuwa i nadpisuje z nowymi informacjami podanymi za pomocą tekstu w stylu HTML. Tekst jest ułożony tak, że na początku pokazuje nam datę i godzinę stworzenia raportu po czym tworzy tabelkę, która w pierwszej kolumnie ma input a w drugiej output. Wraz kolejnymi wierszami pojawiają się liczby wczytane z pliku tekstowego dane jak i output. Na końcu znajduje się podpis autora programu.

```
1 import os
2 from datetime import datetime
3 from os.path import exists
4 l1=[]
5 def otw(file):
6     fil = open(file, "r") //Otworzenie pliku, dodanie elementu do listy
7     for i in fil:
8         l1.append(i)
9     fil.close()
10 def whatever(file):
11     if exists("raport.html"): //Sprawdzenie czy istnieje taki plik
12         os.remove("raport.html") //Usunięcie pliku
13     now=datetime.now()
14     fulldate = now.strftime("%d-%m-%Y %H:%M:%S") //Pobranie aktualnej
        daty i godziny
15     otw(file)
16     fil2=open("raport.html","w") //Otworzenie pliku
17     fil2.write(""" //Ustalenie stylu pliku, tabeli jak i wpisanie danych
        do pliku
18     <!DOCTYPE html>
19     <html>
20         <head>
21             <style>
22                 table, th, td {
23                     border: 1px solid black;
24                 }
25             </style>
26         </head>
27         <body>
28             <h1>Raport z godziny: """+fulldate+"""/h1>
29             <table>
30                 <tr>
31                     <th>Input</th>
32                     <th>Output</th>
33                 </tr>
34                 <tr>
35                     <td>"""+str(l1[0]) +"""/td>
36                     <td rowspan="2">"""+str(l1[2]) +"""/td>
37                 </tr>
38                 <tr>
39                     <td>"""+str(l1[1]) +"""/td>
40                 </tr>
41             </table>
42             <h4>Marcin Ryt</h4>
43         </body>
44     </html>
```

```

45
46     """)
47     fil2.close() // Zamkniecie pliku
48
49 whatever("dane.txt")

```

---

## Zaprzyjzaznione.bat

Program Zaprzyjzaznione.bat to menu wywołujące funkcje. Pierwsza funkcja uruchamia skrypt lp.py i backup.py.

Druga funkcja pokazuje informacje o programie.

Trzecia funkcja kopiuje pliki do folderu o nazwie Backup.

Czwarta funkcja kończy program.

```

1 @echo off
2 :menu
3 cls
4 echo -----Menu-----
5 echo 1. Uruchom program
6 echo 2. Wyszwietl informacje
7 echo 3. Backup
8 echo 4. Zakoncz
9 echo -----
10 echo:
11 set /p wybierz="Wybierz numer polecenia (1,2,3,4) "
12 IF %wybierz%==1 GOTO opcja1
13 IF %wybierz%==2 GOTO opcja2
14 IF %wybierz%==3 GOTO opcja3
15 IF %wybierz%==4 GOTO exit
16 :opcja1
17 cd /d C:\Users\kogut\Desktop\Projekt
18 lp.py
19 backup.py
20 raport.html
21 pause
22 goto menu
23
24 :opcja2
25 echo Program dla zadanych liczb naturalnych sprawdza czy te liczby sa "
    zaprzyjzaznione".
26 echo Liczby zaprzyjzaznione to para liczb ktorych suma dzielnikow
    pierwszej liczby rowna sie drugiej liczbie,
27 echo a suma dzielnikow drugiej liczby rowna sie pierwszej.
28 echo:
29 echo Program zwraca komunikat w zaleznosci czy liczby sa zaprzyjzaznione
    czy tez nie.
30 echo:
31 echo Autor projektu: Marcin Ryt grupa 4/8
32 echo:
33 pause
34 goto menu
35
36 :opcja3

```

```
37 rmdir /s /q Backup
38 mkdir Backup
39 xcopy dane.txt Backup
40 xcopy raport.html Backup
41 pause
42 goto menu
43
44 :exit
```

---