

Języki skryptowe
Liczby zaprzyjaźnione

Joanna Janiszewska
Informatyka, Rok 2 semestr 3
wydział Matematyki Stosowanej
Politechnika Śląska

Gliwice 2017

1. Treść zadania

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.

2. Model matematyczny

Program szuka dzielników wczytanych liczb a i b w zakresie odpowiednio od 1 do $a/2$ i od 1 do $b/2$. Następnie sumuje znalezione dzielniki i porównuje sumę dzielników właściwych liczby a z liczbą b , oraz sumę dzielników właściwych liczby b z liczbą a .

2.1

Przykładowe rozwiązanie I

Weźmy liczby $a=9$ i $b=4$. Dzielnikami właściwymi liczby 9 są liczby 1,3, natomiast dzielnikami właściwymi liczby 4 są liczby 1,2.

$$1+3=4$$

$$1+2=3$$

Tak więc liczby 9 i 4 nie są parą liczb zaprzyjaźnionych.

Przykładowe rozwiązanie II

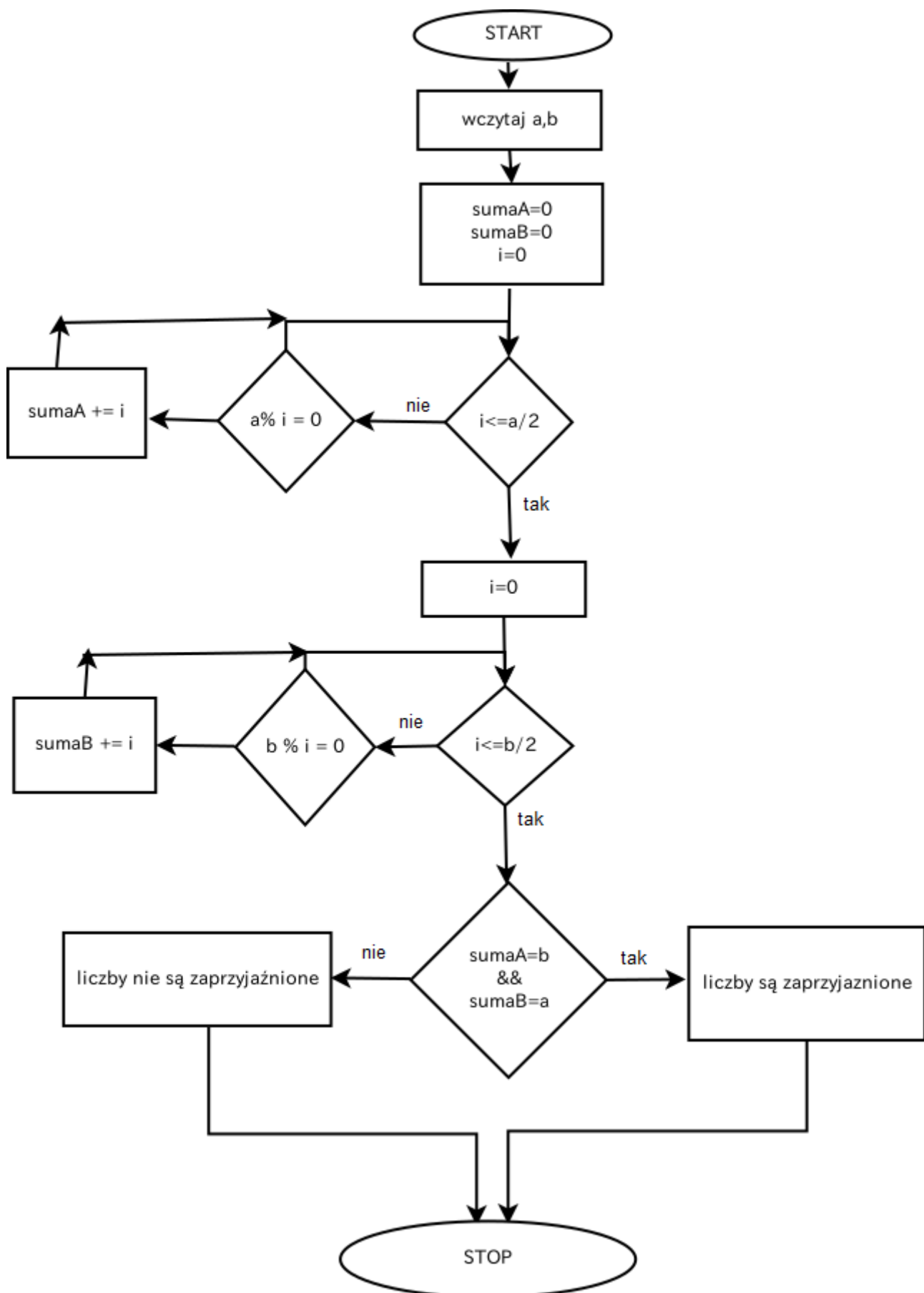
Weźmy liczby: $a=220$ i $b=284$. Dzielnikami właściwymi liczby a są 1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110, a dzielnikami właściwymi liczby b są 1, 2, 4, 71, 142.

$$1 + 2 + 4 + 71 + 142 = 220$$

$$1 + 2 + 4 + 5 + 10 + 11 + 20 + 22 + 44 + 55 + 110 = 284$$

Tak więc liczby 220 i 284 są parą liczb zaprzyjaźnionych.

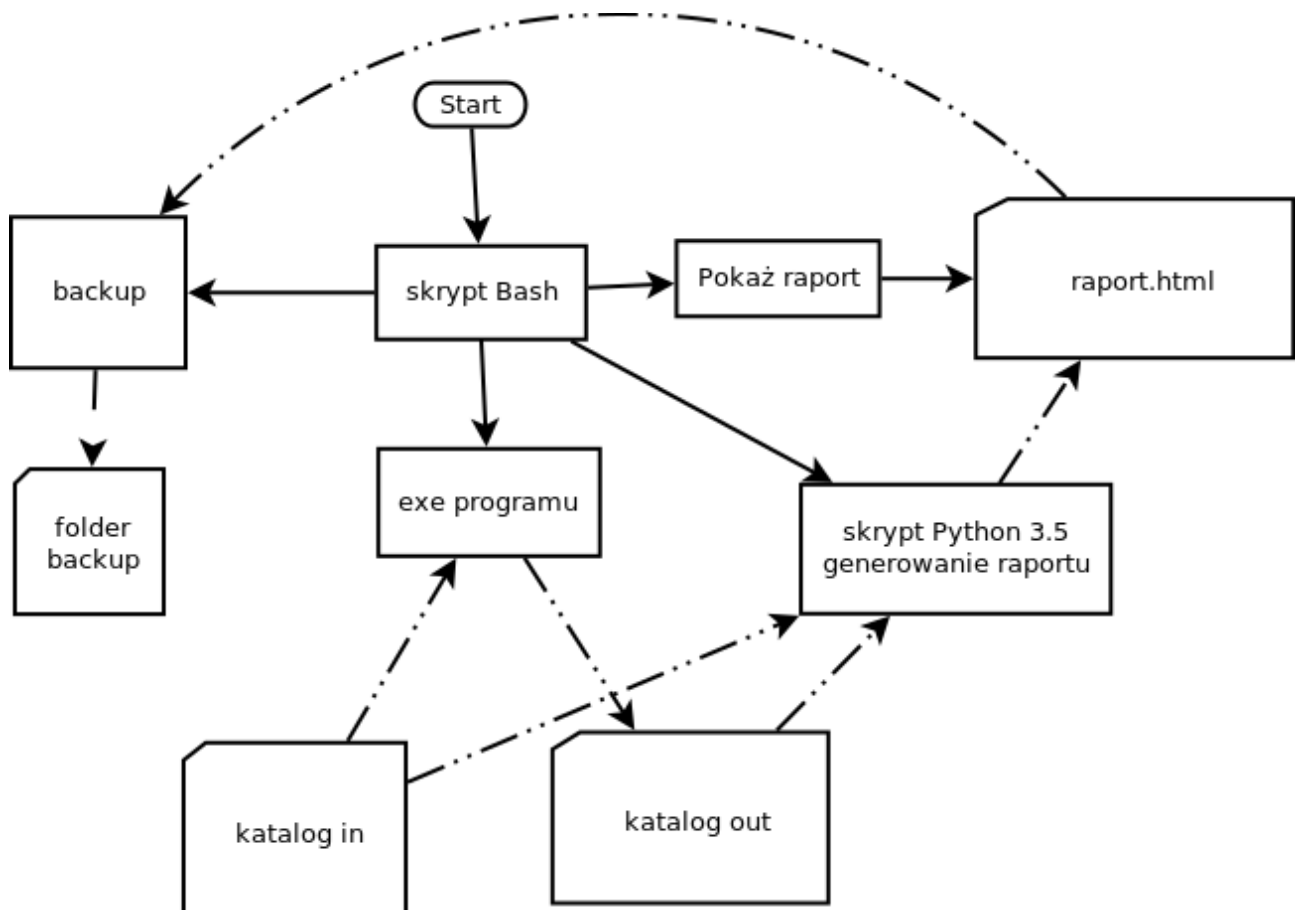
3. Schemat blokowy



4. Pseudokod

1. Wczytaj liczby a,b.
2. $i=0$, sumaA=0, sumaB=0
3. Dopóki $i \leq a/2$
 - 3.1 Jeśli $a \bmod i = 0$ to:
 sumaA+=i
 - 3.2 $i+=1$
4. $i=0$
5. Dopóki $i \leq b/2$
 - 5.1 Jeśli $b \bmod i = 0$ to:
 sumaB+= i
 - 5.2 $i+=1$
6. Jeśli sumaA=b I sumaB=a to liczby są zaprzyjaźnione

5. Implementacja



5.1 Bash:

```
#!/bin/bash
function cpp {
    if test -d "in"; then
        if test ! -d "out"; then
            mkdir "out"
        fi
        pliczki=(./in/*.txt)
        count=${#pliczki[@]}
        for ((number=1; number<=count; number++)); do
            ./liczbyzaprzyjzaznione $(pwd)/in/l$number.txt > ./out/lout$number.txt
        done
        echo "zakonczono przetwarzanie plikow"
    fi
}
function generujraport {
    if test -r "generator.py"; then
        if test -d "in"; then
            if test ! -d "out"; then
                cpp
            fi
            python3 "./generator.py"
            echo "wygenerowano raport"
        fi
    else
        echo "nie mam pliku .py. koncze prace."
    fi
}
function backup {
    if test -r "./raport.html" ; then
        if test ! -d "backup"; then
            mkdir "backup"
        fi
        cp "raport.html" "./backup/backup_`date +%Y_%m_%d_%H_%M_%S`"
        echo "Utworzono backup"
    fi
}
function pokazraport {
    if test -r "./raport.html" ; then
        echo "otwieram raport..."
        xdg-open "raport.html"
    fi
}

function pause(){
    read -p "$*"
}

wybor=666
while [ $wybor -ne 0 ]; do
    case $wybor in
        1)
            # cpp
            cpp
            wybor=666;
            ;;
        2)
            #python
            generujraport
            wybor=666;
            ;;
        3)
            #backup
            backup
    esac
done
```

A screenshot of a Linux desktop environment. The desktop background is a vibrant, abstract image with green, blue, and orange hues. Several desktop icons are visible, including 'Acer', 'zad2.sh', 'LearnOpenGL-master', 'dana_wysciow_e.txt', 'ゴミ箱' (Trash), 'GG' (a smiley face icon), 'ファイルシステム' (File System), '1476991632514.jpg', 'ホーム' (Home), 'xxxx.cpp', 'joannajaska_kold', 'Firefox ウェブブラウザ', 'xxxx.o', 'zad2dana.txt', 'hw.py', and 'スクリーンショット_2017-01-16_19-42-31.png'. A terminal window titled 'Terminal - neb@LinH: ~/Pulpit/projekcik' is open in the center. The terminal shows a menu with options: '1. Uruchom program', '2. Wygeneruj raport', '3. Stworz backup', '4. Otworz istniejacy raport', '0. wyjscie'. The user has selected option 1, and the terminal displays 'Uruchomiono przetwarzanie plikow'. The terminal window has a menu bar with 'ファイル(F)', '編集(E)', '表示(V)', 'ターミナル(T)', 'タブ(A)', and 'ヘルプ(H)'. The taskbar at the bottom shows several open applications: '[Raport - Mozilla Firefox]', '[dokumentacja.odt - LibreOffice]', '[projekcik - ファイルマネージャ]', and 'Terminal - neb@LinH: ~/Pulpit/...'. The system clock in the bottom right corner shows '19 sty, 08:53'.

Zawiera funkcje `cpp()`, `generujraport()`, `backup()`, `pokazraport()`.
`Cpp()` tworzy folder `out` w katalogu projektu, a następnie uruchamia program `c++` dla każdego pliku znajdującego się w folderze `in`. Każdy plik w folderze `in` zawiera parę liczb, a program `c++` określa, czy jest to para liczb zaprzyjaźnionych, czy nie.
Funkcja `generujraport()` uruchamia skrypt `python`, który generuje raport z wynikami programu `C++`. Jeśli w katalogu projektu nie ma folderu `out`, to funkcja `generujraport()` wywołuje funkcję `cpp()`.
Funkcja `backup()` tworzy folder `backup` i kopiuje do niego istniejący już raport. W nazwie pliku kopii zapasowej znajduje się bieżąca data i godzina.
Funkcja `pokazraport()` otwiera istniejący raport w przeglądarce.

5.2 C++

```
#include <iostream>
#include <fstream>
#include <cstdlib>
using namespace std;
bool czy_zaprzyjaznione(int a, int b)
{
    int sumaA=0;
    int sumaB=0;
    for(int i=1 ; i<=a/2 ; i++)
    {
        if(a%i==0)
        {
            sumaA+=i;
        }
    }
    for(int i=1 ; i<=b/2 ; i++)
    {
        if(b%i==0)
        {
            sumaB+=i;
        }
    }

    if(sumaA==b && sumaB==a)
    {
        return true;
    }
    else
    {
        return false;
    }
}
int main(int argc, char *argv[])
{
    int a,b;
    if(argc!=2)
    {
        cout << "Nieprawidlowa ilosc parametrow.\n";
        exit(0);
    }
    else
    {
        ifstream file;
        file.open(argv[1], ios_base::in);

        if (!file.is_open())
        {
            cout << "Wystapil blad. Nie mozna otworzyc pliku.\n";
            exit(1);
        }
        while(!file.eof())
        {
            file >> a;
            file >> b;
            if(czy_zaprzyjaznione(a,b))
            {
                cout << "zaprzyjaznione\n";
            }
            else
            {
                cout << "niezaprzyjaznione\n";
            }
        }
    }
}
```

Program C++ wczytuje plik podany jako argument wejściowy, w którym znajduje się para liczb. Następnie wczytane liczby przekazuje do funkcji czyzaprzyjaznione(), która sprawdza, czy wczytane liczby są zaprzyjaznione. Funkcja zwraca wartość true lub false.

5.3 Python

```
import sys, os, glob
output = open('raport.html', 'w')
output.write( "<!doctype html>\n\
    <html>\n\
    <head>\n\
    <meta charset='utf-8'>\n\
    <link rel='stylesheet' href='style.css' type='text/css'/>\n\
    <title>Raport</title>\n\
    </head>\n\
    <body>\n\
    <h1>Raport</h1>\n\
    <div class='wrap'>"
    )
os.chdir("./in")
number=1
strlp="L.p<br><br>"
strliczby="liczby<br><br>"
str_czyzaprzyjaznione="Zaprzyjaznione?<br><br>"
for i in range(1, len(glob.glob("l*.txt"))+1):
    strlp+=str(i)
    strlp+="<br>"
while number <= len(glob.glob("l*.txt")):
    os.chdir("../in")
    tempIn = open("l"+str(number)+".txt", 'r')
    strg = tempIn.read()
    strg+="<br>"
    strliczby+=strg
    tempIn.close()
    os.chdir("../out")
    tempOut = open("lout"+str(number)+".txt", 'r')
    str_czyzaprzyjaznione+=tempOut.read()
    str_czyzaprzyjaznione+="<br>"
    tempOut.close()
    number+=1
output.write("<div class='lewy'>"+strlp+"</div>")
output.write("<div class='content'> "+strliczby+"</div>")
output.write("<div class='prawy'> "+str_czyzaprzyjaznione+"</div>")
output.write( "</body>\n\
    </html>\n"
    )
output.close()
```




The screenshot shows a web browser window with the title 'Raport - Mozilla Firefox'. The address bar shows the file path 'file:///home/neb/Pulpit/projekcik/raport.html'. The main content area displays a report titled 'Raport' in a large, bold font. Below the title is a table with three columns: 'L.p', 'liczby', and 'Zaprzyjznione?'. The table contains four rows of data. The bottom of the browser window shows the taskbar with several open applications: 'Raport - Mozilla Firefox', 'fdokumentacja.odt - LibreO...', 'projekcik - ファイルマネージャ...', and 'Terminal - neb@Lin: ~/Pulpi...'. The system clock in the bottom right corner indicates '19 sty, 08:52'.

L.p	liczby	Zaprzyjznione?
1	220 284	zaprzyjznione
2	1184 1210	zaprzyjznione
3	202 10	niezaprzyjznione
4	78 67	niezaprzyjznione

6. Podsumowanie

Program sprawdza, czy liczby zawarte w katalogu in są liczbami zaprzyjznionymi. Następnie skrypt python generuje raport z wynikami programu. Skrypt bash w razie potrzeby tworzy backup raportu. Plik style.css odpowiada za estetyczny wygląd raportu.

Program był testowany pod systemami operacyjnymi Windows i Linux. Działa wyłącznie na systemie operacyjnym Linux (32bit i 64bit). Do uruchomienia programu wymagany jest Python w wersji 3.2+.

Projekt można usprawnić poprzez sprawdzanie, czy w katalogu in nie ma duplikatów danych, dodać możliwość posortowania liczb w raporcie.