

Języki Skryptowe

dokumentacja projektu AGENT SMITH

Kinga Osiewicz, grupa 4/7

16 stycznia 2025

Część I

Opis programu

Liczbą Smitha nazywamy taką liczbę naturalną złożoną, której suma cyfr jest równa sumie cyfr wszystkich swoich dzielników pierwszych. Przykładowo liczba 728 jest liczbą Smitha, bo $7 + 2 + 8 = 17$, dodatkowo wiemy, że $728 = 2^3 \times 7 \times 13 = 2 \times 2 \times 2 \times 7 \times 13$ oraz $2 + 2 + 2 + 7 + 1 + 3 = 17$. Napisz program, który weryfikować będzie, czy podana liczba naturalna jest liczbą Smitha. Przykładowo, podając liczbę 728, program napisze, że liczba ta jest liczbą Smitha, a podając jako argument liczbę 777, program napisze, że nie jest to liczba Smitha ($7 + 7 + 7 = 21$, a $777 = 3 \times 7 \times 37$ oraz $3 + 7 + 3 + 7 = 20 \neq 21$). Podobnie będzie dla liczby 787 (choć liczba ta ma sumę cyfr taką jak suma cyfr jej dzielników pierwszych, to nie uznajemy jej za liczbę Smitha, bo nie jest to liczba złożona). Pisząc program, staraj się nie korzystać z dodatkowych bibliotek matematycznych.

Instrukcja obsługi

Aby uruchomić program, należy wykonać następujące kroki:

1. Uruchomić plik MENU.bat.
2. Z menu, wybrać odpowiednią opcję:

```
=====MENU=====
1.  START
2.  INFORMACJE
3.  BACKUP
4.  ZAKONCZ
=====
Wybierz 1,2,3,4:
```

- **START** - Rozpoczyna przetwarzanie plików znajdujących się w katalogu Input oraz tworzy Raport_Smith.html.

```

=====MENU=====
1. START
2. INFORMACJE
3. BACKUP
4. ZAKONCZ
=====
Wybierz 1,2,3,4: 1
Wybrales opcje START
Otwieram plik input1.txt
Otwieram plik input2.txt
Otwieram plik input3.txt
Otwieram plik input4.txt
Raport HTML został wygenerowany.
Press any key to continue . . .

```

- **INFORMACJE** - Wyświetla informacje o liczbach Smitha, ich definicji oraz działaniu programu.

```

=====MENU=====
1. START
2. INFORMACJE
3. BACKUP
4. ZAKONCZ
=====
Wybierz 1,2,3,4: 2
Wybrales opcje INFORMACJE

Liczba Smitha to liczba naturalna zlozona, ktorej suma cyfr jest rowna sumie cyfr wszystkich liczb wystepujacych w jej rozkladzie na czynniki pierwsze. Na przyklad 202 jest liczba Smitha, poniewaz  $2 + 0 + 2 = 4$ , a po jej rozkladzie na czynniki pierwsze  $202 = 2 * 101$  suma cyfr obu czynnikow wynosi  $2 + 1 + 0 + 1 = 4$ . Program weryfikuje, czy podana liczba naturalna jest liczba Smitha. W tym celu wykonywane sa funkcje obliczajace sume cyfr liczby, sprawdzajace, czy liczba jest pierwsza, oraz wyznaczajace wszystkie dzielniki pierwsze. Korzystajac z wynikow poprzednich funkcji, glowna funkcja programu weryfikuje, czy liczba spelnia warunki definicji liczby Smitha: musi byc zlozona, a suma cyfr liczby musi rownac sie sumie cyfr jej dzielnikow pierwszych. Program przetwarza pliki wejsciowe z katalogu "Input", gdzie kazda linia zawiera liczbe do sprawdzenia, i zapisuje wyniki w odpowiednich plikach wyjsciowych w katalogu "Output", informujac, czy liczba jest liczba Smitha.

```

- **BACKUP** - Tworzy kopię zapasową danych wejściowych i wyjściowych, a także pliku raportu Raport_Smith.html. Kopia zapasowa jest zapisywana w folderze backups w podkatalogu nazwanym datą i godziną wykonania kopii.

```

=====MENU=====
1. START
2. INFORMACJE
3. BACKUP
4. ZAKONCZ
=====
Wybierz 1,2,3,4: 3
Wybrales opcje BACKUP
1 file(s) copied.
Kopia zapasowa zostala utworzona w katalogu backups\15.01.2025--13-33-41.

```

- **ZAKONCZ** - Kończy działanie programu.

Podanie innej liczby lub znaku, skutkuje powrotem do menu, aż nie zostanie podana poprawna liczba

3. Program przetwarza pliki znajdujące się w katalogu Input.

4. Program generuje raport w pliku Raport_Smith.html w katalogu Output, w którym informuje, czy dana liczba jest liczbą Smitha.

Wizualizacja danych wejściowych i wyjściowych:

- Pliki wejściowe w katalogu Input zawierają liczby naturalne do sprawdzenia, np.:
 - 202
- Po uruchomieniu programu, dane wyjściowe zostaną zapisane w katalogu Output, np.:
 - Liczba 202 jest liczbą Smitha

Plik raportu Raport_Smith.html zawiera wyniki katalogów

Raport Liczby Smitha - 15-01-2025 13:26:59

Input	Output
728	Liczba 728 jest liczba Smitha.
777	Liczba 777 nie jest liczba Smitha.
202	Liczba 202 jest liczba Smitha.
28	Liczba 28 nie jest liczba Smitha.

- Pliki wejściowe i wyjściowe są także kopiowane do folderu backups w celu tworzenia kopii zapasowej.

0.1 Instrukcja wdrożenia

Dodatkowe informacje

Wymagania:

- Program nie wymaga dodatkowych bibliotek matematycznych.
- Aby prawidłowo uruchomić program, należy mieć zainstalowany Python 3.13 lub wyższy.
- Aby prawidłowo uruchomić program, należy mieć system operacyjny Windows 10 lub nowszy

Podział pracy: Program został podzielony na kilka sekcji:

- **Moduł wejściowy:** Obsługuje pliki wejściowe i ich weryfikację.
- **Moduł obliczeniowy:** Przetwarza dane, wykonując odpowiednie obliczenia.
- **Moduł wyjściowy:** Zapisuje wyniki w odpowiednich plikach wyjściowych.

Wykorzystane biblioteki

W projekcie wykorzystano następujące biblioteki i narzędzia:

- **os** - biblioteka standardowa Pythona umożliwiająca operacje na systemie plików, takie jak tworzenie katalogów czy zarządzanie plikami.
- **datetime** - biblioteka standardowa Pythona używana do pracy z datami i czasem, wykorzystana do generowania znacznika czasu w raporcie.

Biblioteki **os** i **datetime** są częścią standardowej dystrybucji Pythona, co oznacza, że nie wymagają instalacji dodatkowych pakietów. Dzięki nim projekt jest bardziej elastyczny i pozwala na automatyzację wielu operacji plikowych.

Część II

Opis działania

Liczba Smitha to liczba naturalna złożona, której suma cyfr jest równa sumie cyfr wszystkich jej dzielników pierwszych. Liczba n jest liczbą Smitha, jeśli spełnia następujące warunki:

1. Liczba n musi być liczbą złożoną, czyli musi mieć więcej niż dwa dzielniki. Oznacza to, że n nie może być liczbą pierwszą.
2. Suma cyfr liczby n musi być równa sumie cyfr wszystkich jej dzielników pierwszych, uwzględniając ich wielokrotności w rozkładzie na czynniki pierwsze.

Aby obliczyć, czy dana liczba n jest liczbą Smitha, wykonujemy następujące kroki:

- Obliczamy sumę cyfr liczby n . Sumę cyfr liczby n oznaczamy jako $S(n)$.
- Rozkładamy liczbę n na czynniki pierwsze. Rozkład ten zapiszemy w postaci:

$$n = p_1^{k_1} \times p_2^{k_2} \times \cdots \times p_m^{k_m},$$

gdzie p_1, p_2, \dots, p_m to liczby pierwsze, a k_1, k_2, \dots, k_m to ich odpowiednie wykładniki.

- Obliczamy sumę cyfr wszystkich dzielników pierwszych liczby n , uwzględniając ich wystąpienia w rozkładzie na czynniki pierwsze. Jeśli dzielnik p_i występuje k_i razy, to suma cyfr dzielników pierwszych wynosi:

$$S(\text{dzielniki}) = \sum_{i=1}^m k_i \times S(p_i),$$

gdzie $S(p_i)$ to suma cyfr liczby pierwszej p_i .

- Porównujemy sumę cyfr liczby n z sumą cyfr jej dzielników pierwszych. Jeśli:

$$S(n) = S(\text{dzielniki}),$$

to liczba n jest liczbą Smitha. W przeciwnym razie, nie jest liczbą Smitha.

Przykład:

Rozważmy liczbę $n = 728$. Rozkład liczby 728 na czynniki pierwsze wygląda następująco:

$$728 = 2^3 \times 7 \times 13.$$

Suma cyfr liczby 728 wynosi:

$$S(728) = 7 + 2 + 8 = 17.$$

Suma cyfr dzielników pierwszych liczby 728 to:

$$S(\text{dzielniki}) = 3 \times S(2) + 1 \times S(7) + 1 \times S(13) = 3 \times 2 + 1 \times 7 + 1 \times (1 + 3) = 6 + 7 + 4 = 17.$$

Ponieważ suma cyfr liczby 728 jest równa sumie cyfr jej dzielników pierwszych, liczba 728 jest liczbą Smitha.

Algorytmy

Pseudokod

Algorithm 1 Sprawdzanie, czy liczba jest liczbą Smitha oraz obsługa plików

Data: Pliki wejściowe w folderze Input

Result: Pliki wyjściowe w folderze Output

Utwórz folder Output, jeśli nie istnieje **for** $i = 1$ **to** 4 **do**

 Otwórz plik input*i*.txt **if** *plik istnieje* **then**

 Odczytaj liczbę n z pliku **if** *n jest poprawną liczbą całkowitą* **then**

 Oblicz sumę cyfr liczby n Znajdź dzielniki pierwsze liczby n Oblicz sumę cyfr
 dzielników pierwszych **if** *suma cyfr liczby n == suma cyfr dzielników pierwszych*
 then

 | Zapisz do pliku output*i*.txt, że liczba n jest liczbą Smitha

end

else

 | Zapisz do pliku output*i*.txt, że liczba n nie jest liczbą Smitha

end

end

else

 | Wyświetl komunikat, że plik input*i*.txt nie zawiera poprawnej liczby

end

end

else

 | Wyświetl komunikat, że plik input*i*.txt nie istnieje

end

end

Algorithm 2 Obliczanie sumy cyfr liczby

Data: Liczba n

Result: Suma cyfr liczby n

Zamień n na ciąg znaków Zsumuj cyfry jako liczby całkowite Zwróć wynik

Algorithm 3 Sprawdzanie, czy liczba jest pierwsza

Data: Liczba n

Result: True, jeśli n jest pierwsza, w przeciwnym razie False

```
if  $n \leq 1$  then
    | Zwróć False
end
for  $i = 2$  to  $\sqrt{n}$  do
    | if  $n \bmod i = 0$  then
    | | Zwróć False
    | end
end
Zwróć True
```

Algorithm 4 Znajdowanie dzielników pierwszych liczby

Data: Liczba n

Result: Lista dzielników pierwszych liczby n

```
Utwórz pustą listę factors for  $i = 2$  to  $n$  do
    | while  $n \bmod i = 0$  and  $i$  jest liczbą pierwszą do
    | | Dodaj  $i$  do listy factors Podziel  $n$  przez  $i$ 
    | end
end
Zwróć factors
```

Implementacja systemu

Program podzielony jest na trzy główne pliki: main.py, raport.py, oraz MENU.bat. Każdy z nich odpowiada za realizację określonych zadań. Poniżej przedstawiono ich zasadę działania.

1. main.py

Plik main.py zawiera implementację głównych funkcji programu:

- `sum_of_digits(n)`: Oblicza sumę cyfr liczby n .
- `is_prime(n)`: Sprawdza, czy liczba n jest liczbą pierwszą.
- `prime_factors(n)`: Wyznacza dzielniki pierwsze liczby n .
- `is_smith_number(n)`: Weryfikuje, czy n spełnia warunki definicji liczby Smitha.

Listing 1: Fragment kodu main.py

```
1 def sum_of_digits(n):
2     """Funkcja do obliczania sumy cyfr liczby."""
3     return sum(int(digit) for digit in str(n))
4
5 def is_prime(n):
6     """Funkcja do sprawdzania, czy liczba jest pierwsza."""
```



```

7     if n <= 1:
8         return False
9     for i in range(2, int(n**0.5) + 1):
10         if n % i == 0:
11             return False
12     return True
13
14 def prime_factors(n):
15     """Funkcja do znajdowania dzielników pierwszych liczby."""
16     factors = []
17     for i in range(2, n + 1):
18         while n % i == 0 and is_prime(i):
19             factors.append(i)
20             n //= i
21     return factors
22
23 def is_smith_number(n):
24     """Funkcja do sprawdzania, czy liczba jest liczbą Smitha."""
25     if is_prime(n):
26         return False
27     sum_n = sum_of_digits(n)
28     factors = prime_factors(n)
29     sum_factors = sum(sum_of_digits(factor) for factor in factors)
30     return sum_n == sum_factors

```

Główna część programu przetwarza pliki wejściowe z katalogu Input i zapisuje wyniki w katalogu Output.

2. raport.py

Plik raport.py generuje raport w formacie HTML. Raport zawiera zestawienie liczb wejściowych oraz wyników ich analizy.

Listing 2: Fragment kodu raport.py

```

1 from datetime import datetime
2
3 # Tworzenie raportu HTML
4 now = datetime.now()
5 fulldate = now.strftime("%d-%m-%Y %H:%M:%S")
6
7 outputfile = open("Raport_Smith.html", "w")
8 outputfile.write(f"""
9 <html>
10 <head>
11     <title>Raport Liczby Smitha</title>
12 </head>
13 <body>
14 <h1>Raport Liczby Smitha - {fulldate}</h1>
15 <table border="1">
16     <tr>
17         <th>Input</th>
18         <th>Output</th>
19     </tr> """)

```

3. MENU.bat

Skrypt MENU.bat jest interfejsem dla użytkownika. Pozwala na uruchomienie programu, wyświetlenie informacji, utworzenie kopii zapasowej oraz zakończenie działania. Przykład sekcji obsługującej wybór opcji:

Listing 3: Fragment kodu MENU.bat

```
1 :menu
2 cls
3 echo =====MENU=====
4 echo 1.  START
5 echo 2.  INFORMACJE
6 echo 3.  BACKUP
7 echo 4.  ZAKONCZ
8 echo =====
9 set /p select="Wybierz 1,2,3,4: "
10
11 if "%select%"=="1" goto start
12 if "%select%"=="2" goto informacje
13 if "%select%"=="3" goto backup
14 if "%select%"=="4" goto exit
```

Testy

W celu zweryfikowania poprawności działania systemu, przeprowadzono testy dla różnych liczb naturalnych wprowadzonych do programu poprzez pliki wejściowe w katalogu Input. Wyniki obliczeń zapisano w plikach wyjściowych w katalogu Output, a następnie wygenerowano raport w formacie HTML.

Przypadki testowe

Przeanalizowano następujące przypadki testowe:

- Liczby będące liczbami Smitha, np. 202.
- Liczby pierwsze, które zgodnie z definicją nie są liczbami Smitha, np. 101.
- Liczby złożone, które nie spełniają kryteriów liczby Smitha, np. 200.
- Wartości krańcowe, np. 1 (nie jest liczbą złożoną).

Uzyskane wyniki

Wyniki zostały zapisane w plikach wyjściowych, a następnie zaprezentowane w raporcie HTML. Przykładowa analiza wyników:

Liczba wejściowa	Wynik
202	Liczba 202 jest liczbą Smitha.
101	Liczba 101 nie jest liczbą Smitha.
200	Liczba 200 nie jest liczbą Smitha.
1	Liczba 1 nie jest liczbą Smitha.

Tabela 1: Przykładowe wyniki testów programu

Analiza wyników

- Wyniki dla liczb Smitha są poprawne: suma cyfr liczby jest równa sumie cyfr jej dzielników pierwszych.
- Liczby pierwsze zostały poprawnie oznaczone jako nienależące do zbioru liczb Smitha.
- Liczby złożone niespełniające kryteriów definicji liczby Smitha również zostały poprawnie sklasyfikowane.

Wnioski

Przeprowadzone testy potwierdzają poprawność implementacji systemu. Program poprawnie identyfikuje liczby Smitha oraz klasyfikuje liczby naturalne według założeń definicji. Moduł raportujący generuje przejrzysty dokument HTML, co ułatwia analizę wyników.

Pełen kod aplikacji

Listing 4: main.py

```
1 import os
2
3 def sum_of_digits(n):
4     """Funkcja do obliczania sumy cyfr liczby."""
5     return sum(int(digit) for digit in str(n))
6
7 def is_prime(n):
8     """Funkcja do sprawdzania, czy liczba jest pierwsza."""
9     if n <= 1:
10         return False
11     for i in range(2, int(n**0.5) + 1):
12         if n % i == 0:
13             return False
14     return True
15
16 def prime_factors(n):
17     """Funkcja do znajdowania dzielników pierwszych liczby."""
18     factors = []
19     for i in range(2, n + 1):
20         while n % i == 0 and is_prime(i):
21             factors.append(i)
22             n //= i
23     return factors
24
25 def is_smith_number(n):
26     """Funkcja do sprawdzania, czy liczba jest liczbą Smitha."""
27     if is_prime(n):
28         return False # Liczba musi być złożona
29
30     # Suma cyfr liczby n
31     sum_n = sum_of_digits(n)
32
33     # Dzielniki pierwsze
34     factors = prime_factors(n)
35
36     # Suma cyfr dzielników pierwszych
37     sum_factors = sum(sum_of_digits(factor) for factor in factors)
38
39     return sum_n == sum_factors
40
41 # Główna część programu
42 input_dir = "Input"
43 output_dir = "Output"
44 os.makedirs(output_dir, exist_ok=True)
45
46 for i in range(1, 5):
47     input_file = os.path.join(input_dir, f"input{i}.txt")
48     output_file = os.path.join(output_dir, f"output{i}.txt")
49
50     try:
```

```

51         with open(input_file, "r") as infile:
52             number = int(infile.read().strip())
53             result = (f"Liczba {number} jest liczb Smitha." if
54                       is_smith_number(number)
55                       else f"Liczba {number} nie jest liczb Smitha.")
56
57         with open(output_file, "w") as outfile:
58             outfile.write(result)
59
60     except FileNotFoundError:
61         print(f"Plik input{i}.txt nie istnieje.")
62     except ValueError:
63         print(f"Plik input{i}.txt nie zawiera poprawnej liczby.")

```

Listing 5: raport.py

```

1  import os
2  from datetime import datetime
3
4  # Tworzenie raportu HTML
5  now = datetime.now()
6  fulldate = now.strftime("%d-%m-%Y %H:%M:%S")
7
8  outputfile = open("Raport_Smith.html", "w")
9
10 outputfile.write(f"""
11 <html>
12     <head>
13         <title>Raport Liczby Smitha</title>
14     </head>
15     <body>
16         <h1>Raport Liczby Smitha - {fulldate}</h1>
17         <table border="1">
18             <tr>
19                 <th>Input</th>
20                 <th>Output</th>
21             </tr>
22             """
23
24 input_dir = "Input"
25 output_dir = "Output"
26
27 # Generowanie raportu na podstawie plikow wejsciowych i wyjsciowych
28 for i in range(1, 5):
29     input_file = os.path.join(input_dir, f"input{i}.txt")
30     output_file = os.path.join(output_dir, f"output{i}.txt")
31
32     try:
33         with open(input_file, "r") as infile:
34             number = int(infile.read().strip())
35         with open(output_file, "r") as outfile:
36             result = outfile.read().strip()
37
38     # Dodawanie danych do tabeli HTML

```

```

39         outputfile.write(f"""
40         <tr>
41             <td>{number}</td>
42             <td>{result}</td>
43         </tr>
44         """)
45
46     except FileNotFoundError:
47         outputfile.write(f"""
48         <tr>
49             <td>{f"input{i}.txt"}</td>
50             <td>Plik nie istnieje.</td>
51         </tr>
52         """)
53     except ValueError:
54         outputfile.write(f"""
55         <tr>
56             <td>{f"input{i}.txt"}</td>
57             <td>Niepoprawny format liczby.</td>
58         </tr>
59         """)
60
61     outputfile.write("""
62     </table>
63     </body>
64 </html>
65 """)
66
67     outputfile.close()
68
69     print("Raport HTML zostal wygenerowany.")

```

Listing 6: MENU.bat

```

1  @echo off
2  :menu
3  cls
4  echo =====MENU=====
5  echo 1.  START
6  echo 2.  INFORMACJE
7  echo 3.  BACKUP
8  echo 4.  ZAKONCZ
9  echo =====
10 set /p select="Wybierz 1,2,3,4: "
11
12 if "%select%"==" " goto invalid
13 if "%select%"=="1" goto start
14 if "%select%"=="2" goto informacje
15 if "%select%"=="3" goto backup
16 if "%select%"=="4" goto exit
17
18 :invalid
19 echo Wybierz opcje (1-4)
20 pause

```

```

21 goto menu
22
23 :start
24 echo Wybrales opcje START
25
26 REM Przetwarzanie plik w z katalogu Input
27 for /f "delims=" %%a in ('dir /b Input') do (
28     echo Otwieram plik %%a
29     call C:\Users\conta\AppData\Local\Programs\Python\Python313\python.
        exe main.py %%a
30 )
31
32 REM Generowanie raportu
33 call C:\Users\conta\AppData\Local\Programs\Python\Python313\python.exe
    raport.py
34
35 pause
36 goto menu
37
38 :informacje
39 echo Wybrales opcje INFORMACJE
40 echo.
41 echo Liczba Smitha to liczba naturalna zlozona, ktorej suma cyfr jest
    rowna sumie cyfr wszystkich liczb wystepujacych w jej rozkladzie na
    czynniki pierwsze.
42 pause
43 goto menu
44
45 :backup
46 echo Wybrales opcje BACKUP
47 IF NOT EXIST backups mkdir backups
48 set name=%date%--%TIME:~0,2%-%TIME:~3,2%-%TIME:~6,2%
49 set name=%name::-%
50 IF EXIST Raport_Smith.html mkdir backups\%name%
51 robocopy Input backups\%name%\Input /NFL /NDL >nul
52 robocopy Output backups\%name%\Output /NFL /NDL >nul
53 copy Raport_Smith.html backups\%name%\Raport_Smith.html
54 echo Kopia zapasowa zostala utworzona w katalogu backups\%name%.
55 pause
56 goto menu
57
58 :exit
59 echo Wybrales wyjscie
60 pause
61 exit

```
