

# Języki Skryptowe

dokumentacja projektu Zegar 2010

Marceli Pychyński, grupa 3F

14 grudnia 2020

# Spis treści

<b>1</b>	<b>Opis problemu przedstawionego w zadaniu</b>	<b>3</b>
1.1	Treść zadania . . . . .	3
1.2	Założenia . . . . .	3
1.3	Przykładowy format danych wejściowych i wyjściowych . . . . .	3
<b>2</b>	<b>Model Matematyczny</b>	<b>3</b>
2.1	Problem . . . . .	3
2.2	Instrukcja obsługi . . . . .	5
2.3	Wymagania sprzętowe . . . . .	6
<b>3</b>	<b>Algorytm</b>	<b>7</b>
3.1	Pseudokod . . . . .	7
3.2	Schemat blokowy . . . . .	7
<b>4</b>	<b>Implementacja</b>	<b>9</b>
<b>5</b>	<b>Podsumowanie</b>	<b>15</b>
5.1	Co zostało zrobione . . . . .	15
5.2	Pomysły dotyczące rozbudowy . . . . .	15

# 1 Opis problemu przedstawionego w zadaniu

## 1.1 Treść zadania

W zadaniu rozważamy zegar, który wskazuje na godzinę 12:00. Należy obliczyć ile minie sekund, do ponownego ustawienia się wskazówek w jednej linii. Największym problem jest to, że wskazówki pokonują drogę po okręgu w różnym czasie, a wskazówka minutowa „goni”/podąża za wskazówką godzinową. Dalsza część opisu problemu zawarta jest w modelu matematycznym, ponieważ w trakcie rozważań wyprowadzany jest wzór potrzebny do rozwiązania zadania.

## 1.2 Założenia

Wejście - **plik: input.txt**

Pierwszą linią tego pliku jest licznik wyznaczonego w modelu matematycznym pierwiastka. Drugą linią tego pliku jest mianownik ww. pierwiastka.

Wyjście 1 - **plik: output.txt**

Pierwszą linią tego pliku jest obliczony czas w sekundach, a drugą linią tego pliku jest przeliczony czas z sekund na format godzina, minuta, sekunda milisekunda.

Wyjście 2 - **plik: output.html**

Plik zawiera czytelną tabelę stworzoną w html'u z zawartymi wynikami z output.txt

## 1.3 Przykładowy format danych wejściowych i wyjściowych

**input.txt**

12  
11

**output.txt**

3927.272727272727  
1 h 5 m 27 s 27 ms

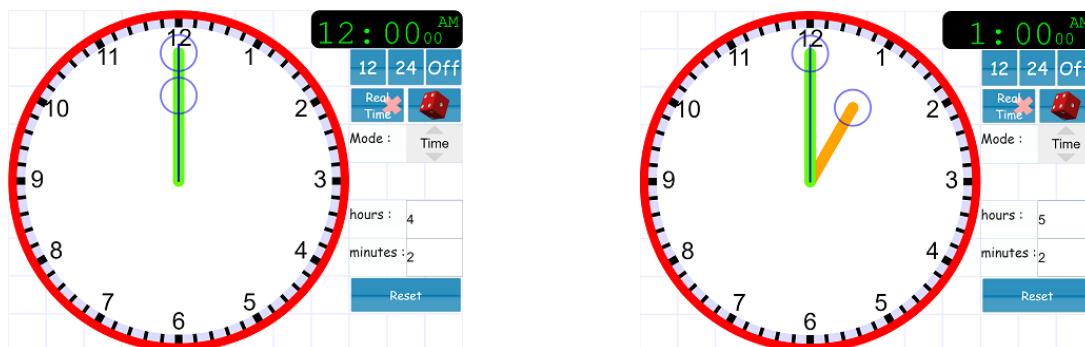
**output.html**

Tabela z danymi wejściowym i wyjściowym

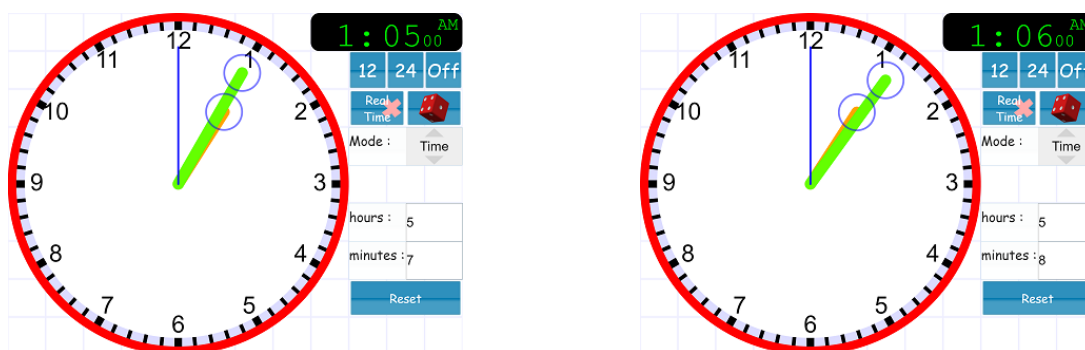
# 2 Model Matematyczny

## 2.1 Problem

W ciągu 12h wskazówka godzinowa pokonuje cały okrąg, zatem w 1h pokona  $1/12$  okręgu, ponieważ na wskazówce minutowej, przejście całego okręgu zajmuje 12h. Z tego wynikają dane (m – minutowa, g – godzinowa):  $t_m = 1h$ ,  $t_g = 12h$



Rysunek 1: Porównanie zegarów w godzinach 12:00 i 13:00



Rysunek 2: Porównanie zegarów w godzinach 13:05 i 13:06

W takim razie, po 1h będzie 13:00. Wskazówki będą od siebie w „odległości” 5 minut (godzinowa na: 1, minutowa na: 0), więc po 5 minutach wskazówka minutowa zbliży się bardzo blisko do wskazówki godzinowej, ale nie znajdą na siebie, ponieważ w ciągu tych 5 minut wskazówka godzinowa pokona 2.5 stopnia (wynika to z prędkości kątowej  $(360^\circ)/12h$ ).

$$s_m \neq s_g$$

Aby wyprowadzić wzór na miejsce spotkania, można do drogi wskazówki godzinowej dodać 2, ponieważ pokonała drogę krótszą o przynajmniej jeden okrąg.

$$s_m = s_g + 2\pi$$

Jednak bez obliczenia, nie jesteśmy w stanie określić dokładnie, na której sekundzie wskazówki ustawią się w jednej linii.

Po krótkiej analizie można uznać, że najlepszą drogą jest tak naprawdę zakreślony kąt (czyli  $s_m = \alpha_m$  i  $s_g = g$ , a takie dane z jakimi się spotkaliśmy można wykorzystać do zastosowania wzoru na prędkość kątową

Ze wzoru na prędkość kątową  $\omega = \frac{\alpha}{t}$ , można wyprowadzić wzór  $\alpha = \omega * t$

$\omega$  – prędkość kątowa

$\alpha$  – zakreślony kąt  
 $t$  – czas pokonania

Teraz zamieniamy  $s_m = s_g + 2\pi$  na:

$$\alpha_m = \alpha_g + 2\pi$$

$\alpha_m = \omega_m t$ , gdzie „ $t$ ” jest szukany czasem, po którym wskazówki się pokrywają

$$\omega_m t = \omega_g t + 2\pi$$

$\omega = \frac{2\pi}{t}$ , a więc:

$$\begin{aligned}\frac{2\pi}{t_m} t &= \frac{2\pi}{t_g} t + 2\pi \\ \frac{t}{t_m} &= \frac{t}{t_g} + 1 \\ t\left(\frac{1}{t_m} - \frac{1}{t_g}\right) &= 1 \\ t &= \frac{1}{\frac{1}{t_m} - \frac{1}{t_g}}\end{aligned}$$

Teraz do wprowadzonego wzoru można podstawić dane, które zostały ustalone na początku rozwiązania:

$$\begin{aligned}t &= \frac{1}{\frac{1}{1} - \frac{1}{12}} \\ t &= \frac{12}{11}(h)\end{aligned}$$

Gdzie  $t$  jest szukany czasem.

## 2.2 Instrukcja obsługi

W celu uruchomienia programów projektowy należy otworzyć plik menu.bat (uprawnienia administracyjne nie są wymagane)



Ładna ikonka skrótu do pliku menu.bat

Projekt zaliczeniowy z języków skryptowych

=====

- 1 Informacja na temat polecenia
- 2 Wykonaj obliczenia (python)
- 3 Utwórz kopię zapasową
- 4 Zakończ

=====

Wybierz opcje (np. 2):2

=====

Od 12:00 do kolejnej godziny, kiedy wskazówki się pokryły minęło: 1 h 5 m 27 s 27 ms

Taki wynik odpowiada: 3927.272727272727 sekundom

Skrypt zadziałał prawidłowo

Zapis raport w postaci pliku html zadziałał prawidłowo

=====

Press any key to continue . . .

## 2.3 Wymagania sprzętowe

System operacyjny Windows 10

Interpreter języka Python w wersji 3.X.X

## 3 Algorytm

### Opis działania

Pseudokod algorytmu:

#### 3.1 Pseudokod

**Data:** Dane wejściowe:  $x, y, var = res = \frac{x}{y}$

**Result:**  $res * 60 * 60$

Wprowadź godzinę w postaci ułamka (np.  $4/3$ ) i zapisz do zmiennej typu rzeczywistego „var” i „res”

Do zmiennej „h” przypisz część całkowitą zmiennej var

Do zmiennej „var” przypisz jako typ rzeczywisty wynik z działania  $(var-h) \times 60$

Do zmiennej „m” przypisz część całkowitą zmiennej var

Do zmiennej „var” przypisz jako typ rzeczywisty wynik z działania  $(var-m) \times 60$

Do zmiennej „s” przypisz część całkowitą zmiennej var

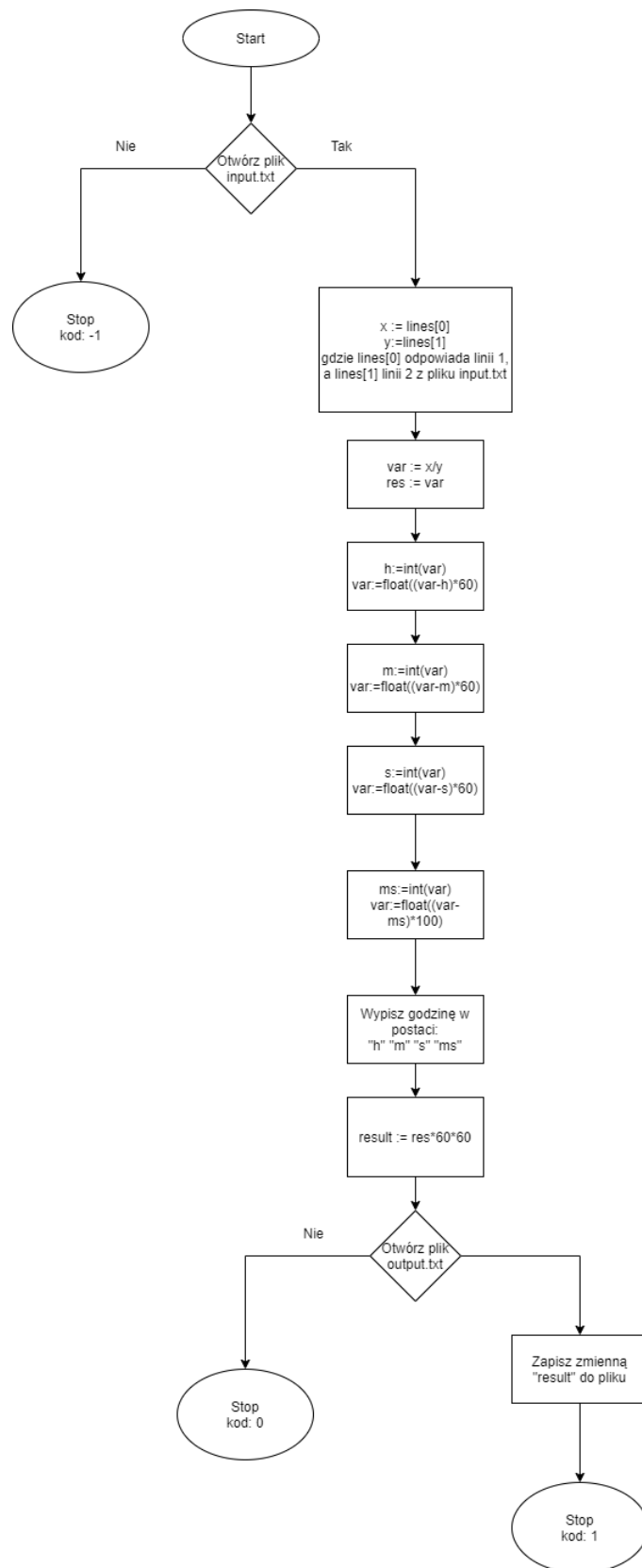
Do zmiennej „var” przypisz jako typ rzeczywisty wynik z działania  $(var-s) \times 100$

Do zmiennej „ms” przypisz część całkowitą zmiennej var

Wypisz godzinę w postaci „h” „m” „s” „ms”

Wypisz ilość sekund używając  $(res*60*60)$

#### 3.2 Schemat blokowy

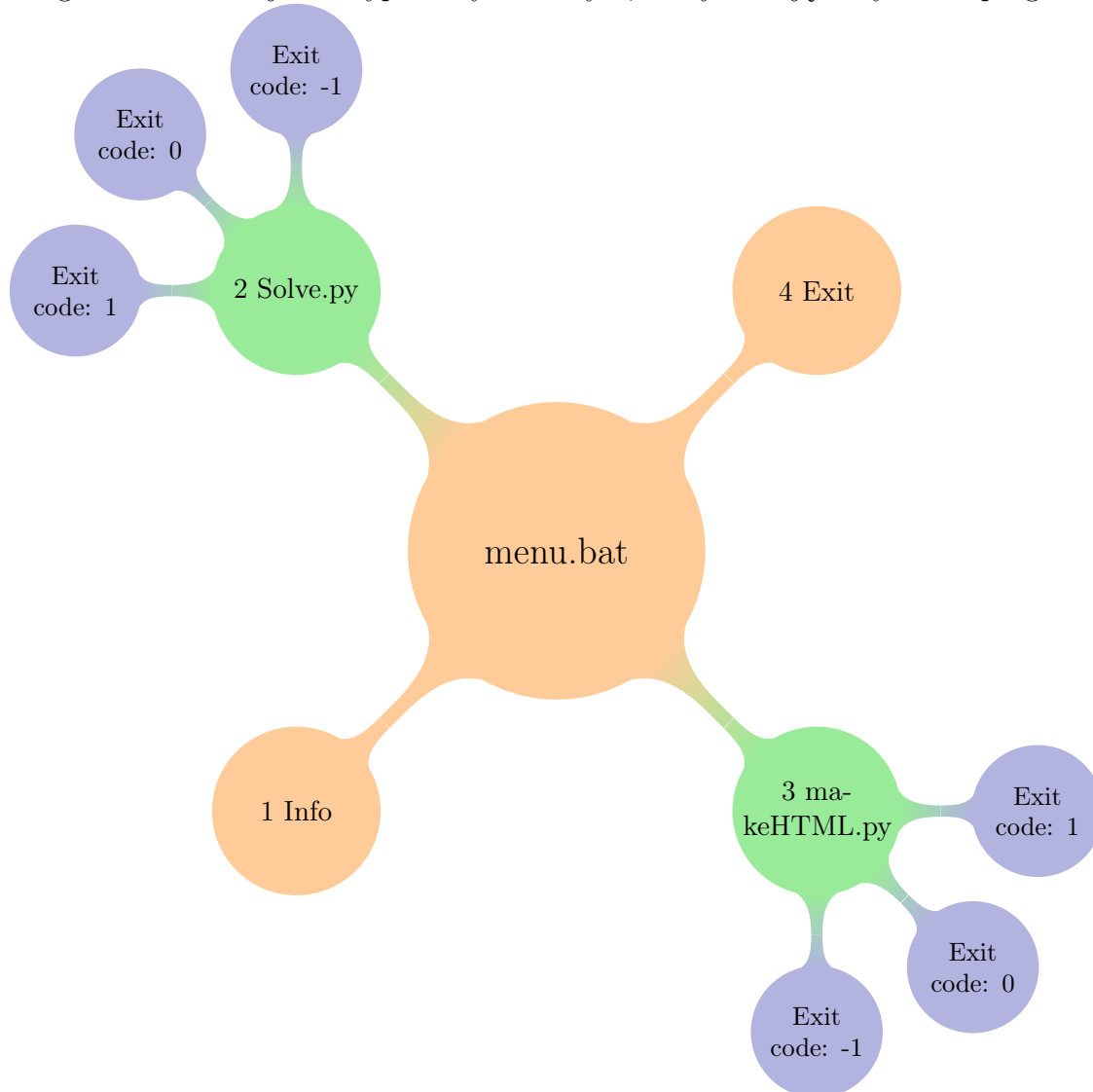




## 4 Implementacja

Opis, zasada i działanie programu ze względu na podział na pliki, następnie funkcje programu wraz ze szczegółowym opisem działania (np.: formie pseudokodu, czy odniesienia do równania)

Program menu.bat jest skryptem systemowym, który steruje wszystkimi programami



Kod wyjścia 1 - skrypt zadziałał prawidłowo

Kod wyjścia 0 - błąd zapisu danych

Kod wyjścia -1 - błąd odczytu danych wejściowych

menu.bat

```
1 @echo off
2 :menu
3 cls
4 echo Projekt zaliczeniowy z języków skryptowych
```

```

5      echo =====
6      echo 1   Informacja na temat polecenia
7      echo 2   Wykonaj obliczenia (python)
8      echo 3   Utworz kop[U+FFFD]zapasowa
9      echo 4   Zakoncz
10     echo =====
11     set /p select=Wybierz opcje (np. 2):
12     IF %select%==1 GOTO opt1
13     IF %select%==2 GOTO opt2
14     IF %select%==3 goto opt3
15     IF %select%==4 goto exit
16     :opt1
17     echo
18     =====
19
18     echo Polecenie jest nastepujace:
19     echo Wskazowki zegara sa ustawione na godzine 1200 . Nalezy napisac
20     echo program
21     echo obliczajacy czas w sekundach jaki uplynal od momentu startu
22     echo zegara do momentu
23     echo ustawienia sie wskazowek w jednej linii. Wynik nalezy zapisac w
24     echo pliku "output.txt".
25     echo
26     =====
27
23     pause
24     goto menu
25     :opt2
26     echo
27     =====
28
27     python solve.py
28     If %ERRORLEVEL% == 1 (echo Skrypt zadzialal prawidlowo)
29     If %ERRORLEVEL% == 0 (echo Blad zapisu plikow wyjsciowych)
30     If %ERRORLEVEL% == -1 (echo Blad otwarcia plikow wejsciowych)
31
32     python makeHTML.py
33     If %ERRORLEVEL% == 1 (echo Zapis raport w postaci pliku html
34     zadzialal prawidlowo)
35     If %ERRORLEVEL% == 0 (echo Blad zapisu plikow wyjsciowych w postaci
36     pliku html)
37     If %ERRORLEVEL% == -1 (echo Blad otwarcia plikow wejsciowych podczas
38     tworzenia raportu)
39     echo
40     =====
41
37
38     pause
39     goto menu
40     :opt3
41     echo
42     =====
43
42     echo Kopiowanie folderu %cd%...
43     echo Usuwanie starej kopii zapasowej...

```

```

44     rmdir /S /Q %userprofile%\Backup
45     echo Tworzenie nowej kopii zapasowej...
46     mkdir %userprofile%\Backup
47     xcopy /e /v "%cd%" "%userprofile%\Backup"
48     echo
=====

49     pause
50     goto menu
51     :exit
52     echo Koniec
53     pause

```

---

solve.py

```

1     lines = []
2
3     try:
4         file_in = open("input.txt", "r")
5         lines = file_in.readlines()
6         file_in.close()
7     except IOError:
8         print("Nie znaleziono pliku")
9         exit(-1)
10
11    try:
12        x = int(lines[0])
13        y = int(lines[1])
14    except ValueError:
15        print("Nieprawidlowe dane wejsciowe, program zostanie wykonany
16              na ponizszych danych:")
17        print("licznik: 12, mianownik: 11")
18        x = 12
19        y = 11
20
21    var = res = float(x / y)
22    h = int(var) # godziny
23    var = float((var - h) * 60)
24    m = int(var) # minuty
25    var = float((var - m) * 60)
26    s = int(var) # sekundy
27    var = float((var - s) * 100)
28    ms = int(var)
29
30    result3 = "{} h {} m {} s {} ms".format(h, m, s, ms)
31    result2 = ("Od 12:00 do kolejnej godziny, kiedy wskazowki sie
32              pokryly minelo: {} h {} m {} s {} ms".format(h, m, s, ms))
33    print(result2)
34    result = float(res * 60 * 60)
35    print("Taki wynik odpowiada: ", result, "sekundom")
36
37    try:
38        file_out = open("output.txt", "w")
39        file_out.writelines(str(result))
40        file_out.writelines("\n")

```

```

39         file_out.writelines(str(result3))
40         file_out.close()
41     except IOError:
42         print("Bład zapisu danych")
43         exit(0)
44
45     exit(1)

```

---

#### makeHTML.py

```

1  # Zapis do pliku html
2  try:
3      file_in = open("input.txt", "r")
4      lines = file_in.readlines()
5      file_in.close()
6  except IOError:
7      print("Nie znaleziono pliku")
8      exit(-1)
9
10 try:
11     x = int(lines[0])
12     y = int(lines[1])
13 except ValueError:
14     print("Nieprawidlowe dane wejsciowe, program zostanie wykonany na
        ponizszych danych:")
15     print("licznik: 12, mianownik: 11")
16     x = 12
17     y = 11
18
19 try:
20     file_in = open("output.txt", "r")
21     lines = file_in.readlines()
22     file_in.close()
23 except IOError:
24     print("Nie znaleziono pliku")
25     exit(-1)
26
27 try:
28     message_res_2 = lines[0]
29     message_res_4 = lines[1]
30 except ValueError:
31     print("Nieprawidlowe dane wejsciowe, program zostanie wykonany na
        ponizszych danych:")
32     print("message_res_2: 3927.272727272727, message_res_4: 1 h 5 m 27 s
        27 ms")
33     message_res_2 = lines[0]
34     message_res_4 = lines[1]
35
36
37 try:
38     file_html = open("output.html", "w")
39
40     message = """
41     <!DOCTYPE html>
42     <html>

```


```

43     <head>
44     <link rel="stylesheet" href="styles.css">
45     </head>
46     <body>
47
48
49     <table class="steelBlueCols">
50     <thead>
51     <tr>
52     <th>Wejscie</th>
53     <th>Wartosc wejscia</th>
54     <th>Wyjscie</th>
55     <th>Wartosc wyjscia</th>
56     </tr>
57     </thead>
58     <tbody>
59     <tr>
60     <td>Licznik</td><td>""
61
62     message_res_1 = x
63     message2 = ""
64     </td>
65     <td>Sekundy</td><td>""
66     #message_res_2 = result
67     message3 = ""
68     </td></tr>
69     <tr>
70     <td>Mianownik</td><td>""
71     message_res_3 = y
72     message4 = ""
73     </td>
74     <td>Czas</td><td>""
75     #message_res_4 = ("{} h {} m {} s {} ms".format(h, m, s, ms))
76     message5 = ""
77     </td></tr>
78     <tr>
79     </tbody>
80     </tr>
81     </table>
82
83     </body>
84     </html>""
85     file_html.writelines(str(message))
86     file_html.writelines(str(message_res_1))
87     file_html.writelines(str(message2))
88     file_html.writelines(str(message_res_2))
89     file_html.writelines(str(message3))
90     file_html.writelines(str(message_res_3))
91     file_html.writelines(str(message4))
92     file_html.writelines(str(message_res_4))
93     file_html.writelines(str(message5))
94     file_html.close()
95 except IOError:
96     print("Blad zapisu danych")
97     exit(0)


```

98

99 `exit(1)`

 C:\Windows\system32\cmd.exe

```
Projekt zaliczeniowy z języków skryptowych
=====
1      Informacja na temat polecenia
2      Wykonaj obliczenia (python)
3      Utworz kopie zapasowa
4      Zakończ
=====
Wybierz opcje (np. 2):1
=====
Polecenie jest następujące:
Wskazówki zegara są ustawione na godzinę 1200 . Należy napisać program
obliczający czas w sekundach jaki upłynął od momentu startu zegara do momentu
ustawienia się wskazówek w jednej linii. Wynik należy zapisać w pliku "output.txt".
=====
Press any key to continue . . .
Projekt zaliczeniowy z języków skryptowych
=====
1      Informacja na temat polecenia
2      Wykonaj obliczenia (python)
3      Utworz kopie zapasowa
4      Zakończ
=====
Wybierz opcje (np. 2):2
=====
Od 12:00 do kolejnej godziny, kiedy wskazowki sie pokryly mineło: 1 h 5 m 27 s 27 ms
Taki wynik odpowiada: 3927.272727272727 sekundom
Skrypt zadziałał prawidłowo
Zapis raport w postaci pliku html zadziałał prawidłowo
=====
Press any key to continue . . .
```

 input — Notatnik

Plik Edycja Format Widok Pomoc

12

11

Lin 2 kol 3

100%

## 5 Podsumowanie

### 5.1 Co zostało zrobione

Program pozwala wczytać wprowadzone dane, potrzebne do obliczenia czasu, po którym wskazówki zegara ustawią się jednej linii. Problem matematyczny nie był, aż tak trudny w tym zadaniu co pozwoliło skupić się na integracji skryptów i utrwalenia wiedzy zdobytej w trakcie nauki języków skryptowych.

### 5.2 Pomysły dotyczące rozbudowy

Ciekawym pomysłem jest stworzenie animacji w pythonie, zależną od danej godziny, minuty i sekundy która pokazywałaby jak wskazówki poruszają się po tarczy zegara.

output.html	Wejście	Wartość wejścia	Wyjście	Wartość wyjścia
	Licznik	12	Sekundy	3927.272727272727
	Mianownik	11	Czas	1 h 5 m 27 s 27 ms