

Języki skryptowe - Python

Lista 3 - grudzień

Punktacja:

- 80-100: bdb
 - 70-79: db+
 - 60-69: db
 - 50-59: dst+
 - 40-49: dst
-

Zadanie 1 (15 pkt)

- (3 pkt) Stwórz moduł, który zawiera niżej wymienione funkcje:
 - konwersja Celsjusz -> Fahrenheit
 - konwersja Fahrenheit -> Celsjusz
 - generowanie N losowych temperatur w Celsjuszach
- (3 pkt) Napisz skrypt, który wykorzystuje powyższy moduł, aby utworzyć plik `celsjusz.txt`, w którym zapisze *n* losowo wygenerowanych temperatur (w Celsjuszach).

*Uwaga: niech *n* będzie pobierane z linii komend; program powinien stosownie reagować, gdy podany przez użytkownika argument nie jest liczbą całkowitą;*

- (4 pkt) Napisz skrypt, który wykorzystuje moduł z pierwszego punktu, aby wczytać plik `celsjusz.txt`, a następnie utworzyć odpowiadający mu plik `fahrenheit.txt`, w którym zapisze przekonwertowane temperatury.
- (5 pkt) Napisz skrypt, który wykorzystuje moduł z pierwszego punktu, aby sprawdzić, czy pliki `celsjusz.txt` i `fahrenheit.txt` zawierają rzeczywiście te same temperatury, ale w innych skalach.

Zadanie 2 (5 pkt)

Napisz program `read.py`, który przyjmuje argumenty z linii komend:

`read.py file [mode]`

gdzie `file` jest argumentem obowiązkowym, a `[mode]` opcjonalnym.

Program powinien drukować na ekranie podany plik w zadanym trybie:

- mode=0 (domyślnie) - drukuje cały plik
- mode=1 - pomija linie zaczynające się od # (komentarze)
- mode=2 - numeruje linie, czyli

1. pierwsza linia z pliku
2. druga linia z pliku
- ...

Uwaga: program powinien zwracać stosowny komunikat, gdy argument obowiązkowy nie zostanie podany

Zadanie 3 (5 pkt)

Plik `studenci_python.txt` zawiera listę studentów uczęszczających na wykład z Pythona.

Plik `studenci_cpp.txt` zawiera listę studentów uczęszczających na wykład z C++.

Napisz skrypt, który wydrukuję listę studentów uczęszczających na oba wykłady.

Wsk: `set.intersection`

Zadanie 4 (5 pkt)

Napisz skrypt, który dla zadanego n generuje n pierwszych wyrazów ciągu “look-and-say”.

Zadanie 5 (20 pkt)

Niech ciastko będzie zadane przez rozmiar S (będący liczbą naturalną).

Niech dziecko będzie zdefiniowane przez łakomstwo L (będące liczbą naturalną), które definiuje minimalny rozmiar ciastka, które zadowoli dziecko (czyli dziecko przyjmie ciastko, gdy $S \geq L$).

Mając dane N ciastek o różnych rozmiarach i M dzieci o różnych łakomstwach, zmaksymalizuj liczbę zadowolonych dzieci. *Jedno dziecko może dostać co najwyżej jedno ciastko!*

Przykład:

Ciastka: [1, 2, 3, 2, 2]

Dzieci: [1, 1, 4, 2, 3, 3]

Maksymalna liczba zadowolonych dzieci: 4

1 dziecko o $L=4$ -> brak odpowiedniego ciastka
 2 dzieci o $L=3$ -> tylko jedno ciastko o $S=3$ (zostają $[1, 2, 2, 2]$)
 1 dziecko o $L=2$ -> zostają $[1, 2, 2]$
 2 dzieci o $L=1$ -> dostają po ciastku i dwa zostają nie użyte

Niech S i L przyjmują wartości o 1 do 10

- (5 pkt) Napisz skrypt do generowania zestawów danych. Program powinien zawierać:

- funkcję, która przyjmuje dwa argumenty (liczbę dzieci M i liczbę ciastek N) i zwraca losowe zestawy S i L ; oba argumenty powinny mieć wartości domyślne; funkcja wywołana bez podania argumentów powinna użyć losowych wartości M i N
- funkcję, która przyjmuje trzy argumenty (dwie listy/krotki i ścieżka do pliku) i dopisuje (w nowej linii) listy/krotki do pliku w formacie

$C1, C2, C3, \dots, CN; L1, L2, L3, \dots, LM$

- funkcję *main*, która pobiera z linii komend dwa argumenty: liczbę zestawów oraz ścieżkę do pliku i wykorzystuje powyższe funkcje, aby utworzyć plik z zestawami danych, np:

```
1, 3, 2, 4; 1, 1, 2, 2
8, 4, 5, 2, 4; 3, 4, 7, 8
...
```

- (5 pkt) Napisz moduł do rozwiązywania wyżej opisanego problemu, który zawiera:

- funkcję, która przyjmuje dwa argumenty (listę ciastek i listę dzieci) i zwraca maksymalną liczbę zadowolonych dzieci
- funkcję *main*, która testuje działanie głównej funkcji z pierwszego podpunktu na kilku przykładach

- (10 pkt) Napisz program do analizy wygenerowanych zestawów. Program powinien wczytywać dane wygenerowane skryptem z pierwszego punktu i wykorzystywać moduł napisany w drugim punkcie. Przykładowy wynik działania programu:

```
Ciastka: [1, 2, 3, 2, 2]
Dzieci: [1, 1, 4, 2, 3, 3]
```

Maksymalna liczba zadowolonych dzieci: 4

```
Ciastka: [1, 1, 1]
Dzieci: [2, 3, 1]
```

Maksymalna liczba zadowolonych dzieci: 1

Zadanie 6 (20 pkt)

Napisz program do zliczania, ile razy podany tekst mieści się na ekranie o zadanej szerokości i wysokości (bez dzielenie wyrazów na dwie linie).

Przykład:

```
liczba kolumn = 10
liczba rzędów = 3
tekst = "Ala ma kota."
```

```
Ala ma****
kota. Ala*
ma kota.**
```

wynik = 2

Program powinien:

- przyjmować 3 argumenty z linii komend: liczba kolumn, rzędów, ścieżka do pliku
- wczytywać tekst z pliku
- drukować tekst na ekranie wypełniając pozostałe spacje wybranym znakiem (jak * w przykładzie powyżej)
- podawać końcowy wynik

Zadanie 7 (20 pkt)

Niech drzewo rośnie wg schematu: wiosną podwaja swoją wysokość, latem rośnie o 1 metr. Napisz program, który oblicza wysokość drzewa mając dane: początkową wysokość, datę zasadzenia, datę końcową (pobierane od użytkownika).

Przykład:

```
data zasadzenia: 01-01-2000
data pomiaru: 01-01-2002
wysokość początkowa: 1 metr
```

```
wiosna 2000: 2 metry
lato 2000: 3 metry
wiosna 2001: 6 metrów
lato 2001: 7 metrów
```

wynik: 7 metrów

Dla zadanych parametrów program powinien:

- (5 pkt) podawać końcową wysokość drzewa
- (5 pkt) zapisywać kolejne zmiany w wysokości w pliku tekstowym
- (5 pkt) posiadać opcję rysowania wykresu zależności wysokości od liczby miesięcy

Dla uproszczenia można przyjąć, że wiosna trwa od początku marca do końca maja, a lato od początku czerwca do końca sierpnia.

W przypadku gdy podana data wypada w środku pory roku:

- (0 pkt) użyj dowolnego założenia
- (5 pkt) dokonaj interpolacji (zakładając liniową zależność)

Zadanie 8 (10 pkt)

Napisz program do gry w kółko i krzyżyk. Przykładowa plansza:

	1	2	3
A			

B	X		

C			O