

---

# PPPD - Lab. 05

*Copyright ©2021 M. Śleszyńska-Nowak i in.*

*Zadanie punktowane, lab 05, grupa A, 2021/2022, autor: Piotr Wolszakiewicz*

---

**Temat: Łącznik liczb nadmiarowych z losowego przedziału**

## Treść zadania

Zadanie polega na stworzeniu programu, który będzie łączył w jedną liczbę, liczby nadmiarowe z wylosowanego przedziału. Program umożliwia wylosowanie początku przedziału oraz końca przedziału. Każdorazowo wyświetla wynik złączenia liczb nadmiarowych, które występują w podanym przedziale w postaci:

`<początek, koniec> => wynik_złączenia_liczb_nadmiarowych`

Liczby nadmiarowe to takie, których suma wszystkich właściwych dzielników tej liczby jest większa niż sama liczba. Pierwszą nadmiarową liczbą jest 12. Posiada ona dzielniki 1, 2, 3, 4 i 6, z sumą równą 16.

Program działa na zasadzie stałej interakcji z użytkownikiem. Mianowicie oczekuje na wprowadzenie akcji, którą użytkownik chce wykonać. Potem podejmuje działanie odpowiadające wybranej akcji, wyświetla wynik i czeka na kolejną akcję. Program kończy działanie po wybraniu akcji odpowiadającej za wyjście. Akcje mogą być wykonywane w dowolnej kolejności. Program powinien być odporny na sytuacje, gdzie nie wszystkie dane są już wypełnione.

Przy inicjalizacji programu losujemy rząd wielkości początku przedziału (`order_of_magnitude`). Z prawdopodobieństwem 60% rząd wielkości powinien wynosić 1, w pozostałych przypadkach 2. Na start początek i koniec przedziału mają być ustawione na `nan` (z biblioteki `math`).

Po każdej z operacji wywołujemy funkcję `concatenate_abundants_from_range` - opisana poniżej oraz wyświetlamy wynik złączenia.

Możliwe akcje to:

- 1 - wylosuj początek przedziału
- 2 - wylosuj koniec przedziału
- 3 - Zapisz wyliczenia do pliku
- 4 - wczytaj stan z pliku
- 5 - wyjdź z programu

Co więcej, program wymaga od nas możliwości zapisu i odczytu stanu aplikacji.

Program ma być podzielony na funkcje, które zostaną ze sobą połączone w `main()`:

- `print_header` - Funkcja wypisuje jaką liczbą odpowiada za jaką akcję. Nic nie zwraca.

- **read\_action** - Funkcja pyta do skutku o numer akcji i sprawdza, czy jest on prawidłowy. Jeśli tak, funkcja zwraca numer akcji jako liczbę całkowitą. W przeciwnym razie wypisany jest komunikat o błędzie, informacja o dostępnych akcjach oraz prośba o wpisanie prawidłowej akcji.
- **generate\_beginning** - Funkcja przyjmuje jako argument rząd wielkości przedziału - **order\_of\_magnitude**. W ramach tej metody rzucamy kostką, o wartościach {1,2,3,4,5,6} - wartości jednakowo prawdopodobne. Rzucamy tyle razy, ile wynosi rząd wielkości. Funkcja zwraca liczbę powstałą ze złączenia wylosowanych wartości. Np. dla 2 rzutów kostką, gdzie wypadło 1 oraz 2, nasza funkcja ma zwrócić liczbę całkowitą 12
- **generate\_end** - Funkcja jako argument również przyjmuje rząd wielkości dla końca przedziału. Powinna zostać wywołana z wartością o 1 większą niż wylosowane na początku **order\_of\_magnitude**. W ramach tej funkcji losowane są cyfry 0-9, przy czym 0 wypada z prawdopodobieństwem 15%, 9 z prawdopodobieństwem 5%, a pozostałe cyfry mają to samo prawdopodobieństwo bycia wylosowanym (10%). Jeśli w ramach losowania wylosowaliśmy 3 cyfry np. 3, 9 i 0 to powinniśmy zwrócić liczbę całkowitą 390. Zwracana liczba musi mieć zadany rząd wielkości. Jeśli jako pierwsza losowana cyfra wypadnie 0, to losowanie pierwszej cyfry powtarzamy aż do skutku.
- **concatenate\_abundants\_from\_range** - Funkcja dostaje jako argument początek i koniec przedziału. Dla wszystkich liczb w danym przedziale (włączając brzegi), sprawdzane jest, czy liczba jest nadmiarowa. Funkcja jako wynik zwraca liczbę powstałą przez połączenie wszystkich liczb nadmiarowych w kolejności rosnącej z zadanego przedziału.  
Np. dla przedziału 10 - 24 liczbami nadmiarowymi są 12, 18, 20 i 24. Dla takiego przedziału powinniśmy otrzymać liczbę 12182024. W ramach tej funkcji sprawdzamy, czy początek i koniec przedziału ma już ustawioną wartość, jeśli nie funkcja zwraca jako rezultat **math.nan**
- **save\_file** - Funkcja przyjmuje jako parametry początek i koniec przedziału, a także wynik konkatencji i rząd wielkości. Funkcja pyta użytkownika o ścieżkę do pliku, gdzie stan programu ma być zapisany, a następnie zapisuje ten stan. Funkcja nic nie zwraca. Jeśli początek bądź koniec przedziału jest niewypełniony, funkcja ma wypisać odpowiedni komunikat "Wymagane dane nie są wypełnione" i nic nie zapisywać
- **read\_from\_file** - Funkcja pyta użytkownika o ścieżkę do pliku, gdzie stan aplikacji został zapisany. Funkcja zwraca jako krotkę początek i koniec przedziału, a także wynik konkatencji liczb nadmiarowych i rząd wielkości.

Do sprawdzania, czy zmienna przyjmuje wartość **nan** można użyć funkcji **isnan** z biblioteki **math**:

```
import math
zmienna = math.nan
math.isnan(zmienna)
```

Jeśli został już wylosowany początek bądź koniec przedziału, a ponownie wybieramy akcję do losowania dla niego, to nowa wartość nadpisuje starą. W takim wypadku powinien się wyświetlić nowy wynik konkatencji dla nowego przedziału.

## Ograniczenia/ustawienia startowe

- Proszę ustawić **seed** dla funkcji **random** wartością 2014
- Nie wolno korzystać z tablic
- Do generowania złączeń liczb należy wykorzystać operacje całkowitoliczbowe (nie wolno korzystać z typu **str**)

---

## Przykłady interakcji użytkownika z programem

```
Początek przedziału będzie 1-cyfrowy
Wybierz opcję co chcesz zrobić:
1 - wylosuj początek przedziału
2 - wylosuj koniec przedziału
3 - Zapisz operację do pliku
4 - wczytaj operację z pliku
5 - wyjdź z programu
Podaj akcję którą chcesz wykonać: 1
< 4 : nan > => nan
Podaj akcję którą chcesz wykonać: 1
< 2 : nan > => nan
Podaj akcję którą chcesz wykonać: 0
Wprowadziłeś nieprawidłową wartość: 0
Wybierz opcję co chcesz zrobić:
1 - wylosuj początek przedziału
2 - wylosuj koniec przedziału
3 - Zapisz operację do pliku
4 - wczytaj operację z pliku
5 - wyjdź z programu
Podaj akcję którą chcesz wykonać: 1
< 3 : nan > => nan
Podaj akcję którą chcesz wykonać: 3
< 3 : nan > => nan
Wszystkie pozycje muszą być wypełnione aby zapisać je do pliku
Podaj akcję którą chcesz wykonać: 2
< 3 : 83 > => 1218202430364042485456606670727880
Podaj akcję którą chcesz wykonać: 2
< 3 : 60 > => 121820243036404248545660
Podaj akcję którą chcesz wykonać: 2
< 3 : 30 > => 1218202430
Podaj akcję którą chcesz wykonać: 3
< 3 : 30 > => 1218202430
Podaj nazwę pliku do zapisu: stan_aplikacji.txt
Dane zostały zapisane do pliku: stan_aplikacji.txt
Podaj akcję którą chcesz wykonać: 2
< 3 : 69 > => 12182024303640424854566066
Podaj akcję którą chcesz wykonać: 2
< 3 : 34 > => 1218202430
Podaj akcję którą chcesz wykonać: 2
< 3 : 20 > => 121820
Podaj akcję którą chcesz wykonać: 2
< 3 : 11 > => 0
Podaj akcję którą chcesz wykonać: 4
< 3 : 11 > => 0
Podaj nazwę pliku do odczytu: stan_aplikacji.txt
< 3 : 30 > => 1218202430
Podaj akcję którą chcesz wykonać: 2
< 3 : 23 > => 121820
Podaj akcję którą chcesz wykonać: 5
< 3 : 23 > => 121820
Wyjście
```

---

## Punktacja

Za poszczególne elementy można uzyskać następującą liczbę punktów:

- Prawidłowo stworzony rdzeń programu: obsługa wczytania akcji, wyświetlania informacji na temat dostępnych akcji, zakończenia działania programu, dopiero gdy wybierzemy odpowiednią akcję - **2pkt**
- Poprawnie zaimplementowana logika związana z losowaniem: rzędu wielkości i początku przedziału - **1pkt**
- Poprawnie zaimplementowana logika związana z losowaniem końca przedziału.  
Ręczne sprawdzenie wszystkich warunków - **1pkt**  
Rozwiązanie sprytniejsze - **2pkt**
- Zaimplementowana funkcja `concatenate_abundants_from_range` z obsługą przypadku dla nieustawionych argumentów wejściowych. Wymagane jest, aby wynik tej funkcji był automatycznie wyświetlany po każdym losowaniu/zmianie przedziału - **2pkt**
- Obsługa zapisu stanu aplikacji do pliku - **1pkt**
- Obsługa odczytu stanu aplikacji z pliku - **2pkt**
- Każdy z wymienionych etapów wymaga, aby były stworzone odpowiednie funkcje do tego etapu i aby były one prawidłowo wywołane w `main`

## Uwaga

- Jeśli program się nie kompiluje (interpretuje), ocena jest zmniejszana o połowę
- Jeśli kod programu jest niskiej jakości (nieestetycznie formatowanie, mylące nazwy zmiennych itp.), ocena jest zmniejszana o **2pkt**