

Wstęp do programowania

Pracownia 4

Uwaga: Na tej liście też będą wprawki. Obowiązuje promocja, dzięki której zadania z drugiej listy jeszcze będą warte 1 punkt, a nie 0.5. Niemniej na kolejnych zajęciach już nie będzie można ich oddawać. Dodatkowo, jeżeli nie zrobiłeś zadania ze smokiem z pierwszej listy, możesz je oddać na tych zajęciach (za 1p). Lista 3 ciągle jest warta 100%. Premia 0.5 za tę listę jest za zdobycie co najmniej 2.5 punktu. Premia ta wlicza się do maksimum.

Zadanie 1.(2pkt) Na stronie wykładu jest program, który wykonuje pewne operacje na listach. Zmodyfikuj go zgodnie z zawartymi w nim poleceniami. Oczywiście oddając program musisz go w pełni rozumieć, również te fragmenty, których nie jesteś autorem. **Uwaga:** to jest bardzo proste zadanie i nikt nie powinien go pominąć.

Zadanie 2.(1pkt) Napisz funkcję `randperm(n)` która losuje permutację liczb od 0 do $n - 1$ (czyli ma zwracać wylosowaną listę o długości n zawierającą wszystkie liczby z zadanego przedziału). Przygotuj prezentację tej funkcji, która wypisuje kilka losowych permutacji dla liczb z przedziału od 0 do 9. Nie wolno korzystać z funkcji `shuffle` z modułu `random`.

Za to zadanie można dostać premię 0.5p (nie wliczającą się do maksimum), jeżeli funkcja `randperm` będzie poprawna, będzie losować każdą permutację z tym samym prawdopodobieństwem, a ponadto `x = randperm(10 ** 6)` wykona się w kilka sekund.

Zadanie 3.(1pkt) W zadaniu tym będziemy analizować zadanie ze smokiem. Masz napisać funkcję `analizaSmoka(n)`, która przeprowadza n pełnych rozgrywek „gry ze smokiem” (każda rozgrywka składa się z k serii po 100 rzutów każda), zapisując liczbę serii, które musiały się odbyć do momentu, kiedy smok nas wypuścił.

Po skończonej symulacji funkcja powinna wypisać raport w następującym formacie:

```
Liczba gier: 1000
Najkrótsza gra: 1
Najdłuższa gra: 772
Średnia gra: 97.04
Mediana: 66
```

Liczby powyżej są przykładowym wynikiem wywołania `analizaSmoka(1000)`, Twoje wyniki zapewne będą inne. Jeżeli będą różniły się znacznie w kolejnych wywołaniach, oznacza to najprawdopodobniej, że musisz przemyśleć poprawność Twojej implementacji.

Oczywiście pojedyncza rozgrywka powinna być zamknięta w osobnej funkcji.