
PPPD - Lab. 05

Copyright ©2021 M. Śleszyńska-Nowak i in.

Zadanie punktowane, lab 05, 2020/2021, autor: Małgorzata Śleszyńska-Nowak

Temat: Gra w kości

Treść zadania

Zasady gry:

- Gra dwuosobowa, gracze rzucają kośćmi na zmianę.
- Na początku gry losowana jest liczba wygrywająca (od 10 do 50).
- W każdym ruchu gracz wybiera iloma kośćmi chce rzucić: 1, 2 lub 3. Na każdej kości wypada 1, 2, 3, 4, 5 lub 6 oczek.
- Po każdym ruchu gracz dostaje bonusowe punkty: niech n będzie równe sumie wyrzuconych w danej turze oczek, wówczas liczba bonusowych punktów równa jest liczbie dzielników z zakresu $[2, n-1]$ liczby n .
- Dla każdego gracza sumujemy liczbę oczek, które wyrzucił podczas całej rozgrywki.
- Gra kończy się po turze, w której suma oczek wyrzuconych przez któregoś z graczy równa jest co najmniej liczbie wygrywającej.
- Wygrywa gracz, którego wynik jest bliżej liczby wygrywającej (nie musi to być gracz z większą liczbą punktów).
- Jeżeli wyniki obu graczy są tak samo blisko liczby wygrywającej, gra kończy się remisem.

Napisz program obsługujący wyżej opisaną grę.

Ziarno generatora liczb losowych ustaw na 1354.

Program musi mieć zdefiniowane funkcje:

- `rzucaj` - przyjmuje liczbę kości do rzutu, wypisuje na konsoli informacje o wyrzuconych oczkach, zwraca sumę wyrzuconych oczek

UWAGA: przy zapisie do pliku funkcja potrzebuje dodatkowego parametru

- `sprawdz_bonus` - przyjmuje sumę wyrzuconych oczek i zwraca liczbę bonusowych punktów

cała logika rozgrywki może być zaimplementowana w funkcji `main()`.

Przebieg gry należy zapisać do pliku w postaci:

```
Liczba wygrywająca: 44
Gracz 1 liczba kości: 3 wyrzucone: 1 2 2 bonus: 0 suma: 5
Gracz 2 liczba kości: 3 wyrzucone: 2 3 2 bonus: 0 suma: 7
Gracz 1 liczba kości: 2 wyrzucone: 4 5 bonus: 1 suma: 15
Gracz 2 liczba kości: 2 wyrzucone: 4 5 bonus: 1 suma: 17
Gracz 1 liczba kości: 3 wyrzucone: 5 2 5 bonus: 4 suma: 31
Gracz 2 liczba kości: 3 wyrzucone: 4 1 2 bonus: 0 suma: 24
Gracz 1 liczba kości: 3 wyrzucone: 2 6 4 bonus: 4 suma: 47
Gracz 2 liczba kości: 3 wyrzucone: 4 6 3 bonus: 0 suma: 37
WYGRYWA Gracz 1!
```

Wydruk z konsoli dla powyższego przykładu:

```
Liczba wygrywająca: 44
Suma gracza 1: 0
Suma gracza 2: 0
Ruch gracza 1.
Iloma kośćmi chcesz rzucić?
3
Wyrzuciłeś 1
Wyrzuciłeś 2
Wyrzuciłeś 2
Bonus: 0

Liczba wygrywająca: 44
Suma gracza 1: 5
Suma gracza 2: 0
Ruch gracza 2.
Iloma kośćmi chcesz rzucić?
3
Wyrzuciłeś 2
Wyrzuciłeś 3
Wyrzuciłeś 2
Bonus: 0

Liczba wygrywająca: 44
Suma gracza 1: 5
Suma gracza 2: 7
Ruch gracza 1.
Iloma kośćmi chcesz rzucić?
2
Wyrzuciłeś 4
Wyrzuciłeś 5
Bonus: 1

Liczba wygrywająca: 44
Suma gracza 1: 15
Suma gracza 2: 7
Ruch gracza 2.
Iloma kośćmi chcesz rzucić?
2
Wyrzuciłeś 4
Wyrzuciłeś 5
Bonus: 1

Liczba wygrywająca: 44
Suma gracza 1: 15
Suma gracza 2: 17
Ruch gracza 1.
Iloma kośćmi chcesz rzucić?
3
Wyrzuciłeś 5
Wyrzuciłeś 2
Wyrzuciłeś 5
Bonus: 4
```

```
Liczba wygrywająca: 44
Suma gracza 1: 31
Suma gracza 2: 17
Ruch gracza 2.
Ilość kości chcesz rzucić?
3
Wyrzuciłeś 4
Wyrzuciłeś 1
Wyrzuciłeś 2
Bonus: 0

Liczba wygrywająca: 44
Suma gracza 1: 31
Suma gracza 2: 24
Ruch gracza 1.
Ilość kości chcesz rzucić?
3
Wyrzuciłeś 2
Wyrzuciłeś 6
Wyrzuciłeś 4
Bonus: 4

Liczba wygrywająca: 44
Suma gracza 1: 47
Suma gracza 2: 24
Ruch gracza 2.
Ilość kości chcesz rzucić?
3
Wyrzuciłeś 4
Wyrzuciłeś 6
Wyrzuciłeś 3
Bonus: 0

Liczba wygrywająca: 44
Suma gracza 1: 47
Suma gracza 2: 37
WYGRYWA Gracz 1!

Process finished with exit code 0
```

Punktacja

Za poszczególne elementy można uzyskać następującą liczbę punktów:

- Logika gry (naprzemienny ruch obu graczy, wczytanie liczby kości do rzutu, wyrzucenie wyjątku przy niepoprawnych danych, wypisywanie przebiegu gry na konsolę) - 2p
- Funkcja rzucaj (bez obsługi zapisu do pliku) wraz z poprawnym sumowaniem wyrzuconych przez obu graczy oczek - 2p
- Funkcja sprawdz_bonus wraz z poprawnym sumowaniem bonusowych punktów obu graczy - 1p
- Obsługa końca gry (wypisanie informacji o wygranej na konsoli) - 1p
- Poprawne zapisanie całej rozgrywki do pliku - 4p

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 2 p.