

Nazwisko i imię: _____ Nr indeksu: _____

MiNI/MatLic/AiPP/2014–2015/Kolokwium-IIA

1	2	3	4	Σ
				(30)

GRUPA ĆWICZENIOWA (ZAKREŚL ODPOWIEDNIĄ): MG8 MG13 MB13 MD13 BT13

Uwaga: Za każde zadanie można uzyskać tę samą liczbę punktów. Do ostatecznej oceny wliczane są punkty za 3 najlepiej rozwiązane zadania. Innymi słowy, jednego zadania z poniższych wcale nie musisz (choć możesz) rozwiązywać. Powodzenia!

Zadanie IIA1. Napisz funkcję `DuzyLotek()`, która zasymuluje losowania Dużego Lotka. Przyjmuje ona jako argument tablicę 6 różnych liczb całkowitych z zakresu od 1 do 49 (ułożonych w dowolnej kolejności; faktu, że liczby są różne, nie musisz sprawdzać), reprezentującą liczby typowane przez gracza (czyli jego/jej kupon) oraz dodatnią liczbę całkowitą k . Funkcja ta ma symulować k losowań wspomnianej wyżej loterii – w każdej z k iteracji generowanych jest 6 *różnych* liczb całkowitych z zakresu od 1 do 49. Jako wynik zwracana jest k -elementowa tablica liczb całkowitych, której elementy zawierają informację, ile liczb z kuponu znalazło się w kolejno generowanych wygrywających szóstkach. Dla przykładu, załóżmy, że gracz typuje 1, 6, 9, 49, 43, 29 oraz że decyduje się grać dwukrotnie. W pierwszym losowaniu wylosowane zostały liczby 1, 2, 3, 4, 5, 6, a w drugim – 1, 8, 17, 29, 43, 44. Oczekiwany wynik powinien być zatem tablica (2, 3).

Wskazówka: Przydatna może być funkcja `rand()`, która generuje liczby całkowite z przedziału od 0 do `RAND_MAX-1` włącznie. Uważaj na powtórzenia liczb przy losowaniu (przykładowo, możesz generować liczby dopóki nie znajdziesz takiej, która się jeszcze nie pojawiła).

Kod rozwiązania wraz z komentarzami:

Zadanie IIA2. Wprowadźmy pojęcie macierzy diagonalno dodatnio-ujemnej. Macierz ta ma same dodatnie elementy, które spełniają warunek: $\{a_{ij} : i + j = k, k \text{ jest parzyste}\}$ i same ujemne elementy: $\{a_{ij} : i + j = k, k \text{ jest nieparzyste}\}$ lub na odwrót: macierz ta ma same ujemne elementy, które spełniają warunek: $\{a_{ij} : i + j = k, k \text{ jest parzyste}\}$ i same dodatnie elementy: $\{a_{ij} : i + j = k, k \text{ jest nieparzyste}\}$, gdzie a_{ij} to element leżący na przecięciu i -tego wiersza i j -tej kolumny.

Napisz funkcję `plusminus()`, która zwraca wartość logiczną `true`, gdy dana macierz całkowitoliczbowa (niekoniecznie kwadratowa) M jest dodatnio-ujemna bądź `false` w przeciwnym wypadku.

Przykładowe macierze dodatnio-ujemne:

$$M_1 = \begin{bmatrix} 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad M_2 = \begin{bmatrix} -1 & 2 & -3 \\ 4 & -5 & 6 \\ -3 & 2 & -1 \end{bmatrix}$$

Kod rozwiązania wraz z komentarzami:

Zadanie IIA3. Wyobraźmy sobie, że żyjemy w krainie, w której każdy dzień może mieć inną liczbę godzin, a każda z godzin może mieć inną liczbę minut.

- a) Zdefiniuj strukturę `Dzien`, która zawiera informację o tym, ile jest godzin w danym dniu (liczba całkowita) i ile minut ma każda z tych godzin (tablica liczb całkowitych).
- b) Napisz funkcję `stworzDzien()` (przyjmującą dwa argumenty: liczbę godzin i liczbę minut), która stworzy strukturę `Dzien` odpowiadającą dniu o podanej liczbie godzin, w którym każda godzina ma tyle samo minut (szczególny przykład dla naszego „mikroświata”).
- c) Napisz funkcję `czyPoprawny()` (przyjmującą dowolny obiekt typu `Dzien` – niekoniecznie wygenerowany przez `stworzDzien()` – oraz aktualny czas, tzn. godzinę i minuty – dwie liczby całkowite), która sprawdzi, czy podany czas jest poprawny w podanym dniu.

Przykład: godzina „szesnasta sześćdziesiąt dziewięć” jest niepoprawna w dniu, w którym szesnasta godzina ma mniej niż siedemdziesiąt minut, lub w którym jest mniej niż siedemnaście godzin. Uwaga: godziny i minuty numeruj zaczynając od 0 (jak w życiu, pierwsza minuta pierwszej godziny to 0:00).

Kod rozwiązania i komentarze:

Zadanie IIA4. Napisz funkcję `rzadko()`, która dla danego napisu (zakładamy, że występują w nim tylko wielkie litery alfabetu łacińskiego – tego warunku nie musisz sprawdzać) zwróci pojedynczy znak (`char`), który w napisie wejściowym występuje najrzadziej (np. dla `"ABAACAB"` będzie to `'C'`) – w przypadku niejednoznaczności zwróć dowolną wartość spełniającą tę własność.

Kod rozwiązania i komentarze:

Nazwisko i imię: _____ Nr indeksu: _____

MiNI/MatLic/AiPP/2014–2015/Kolokwium-IIB

1	2	3	4	Σ
				(30)

GRUPA ĆWICZENIOWA (ZAKREŚL ODPOWIEDNIĄ): MG8 MG13 MB13 MD13 BT13

Uwaga: Za każde zadanie można uzyskać tę samą liczbę punktów. Do ostatecznej oceny wliczane są punkty za 3 najlepiej rozwiązane zadania. Innymi słowy, jednego zadania z poniższych wcale nie musisz (choć możesz) rozwiązywać. Powodzenia!

Zadanie IIB1. Wprowadźmy pojęcie macierzy diagonalno parzysto-nieparzystej. Rozważmy rodzinę zbiorów daną wyrażeniem: $A_k = \{a_{ij} : i + j = k\}$ (a_{ij} to element leżący na przecięciu i -tego wiersza i j -tej kolumny; zbiór ten to taka specyficzna „diagonała” idąca od lewego dolnego rogu do prawego górnego). Macierz jest macierzą diagonalno parzysto-nieparzystą, gdy dla każdego parzystego k elementy ze zbioru A_k są liczbami parzystymi, a dla każdego nieparzystego k elementy ze zbioru A_k są liczbami nieparzystymi albo jest dokładnie na odwrót: dla każdego parzystego k elementy ze zbioru A_k są liczbami nieparzystymi, a dla każdego nieparzystego k elementy ze zbioru A_k są parzyste.

Napisz funkcję `parznieparz()`, która zwraca wartość logiczną `true`, gdy dana macierz całkowitoliczbowa (niekoniecznie kwadratowa) M jest parzysto-nieparzystą bądź `false` w przeciwnym wypadku. Rozważamy tylko macierze, które mają elementy nieujemne. Przykładowe macierze diagonalno parzysto-nieparzyste:

$$M_1 = \begin{bmatrix} 1 & 2 & 1 & 2 \\ 2 & 1 & 2 & 1 \\ 1 & 2 & 1 & 2 \end{bmatrix} \quad M_2 = \begin{bmatrix} 2 & 3 & 4 \\ 5 & 8 & 1 \\ 0 & 3 & 6 \end{bmatrix}$$

Kod rozwiązania wraz z komentarzami:

Zadanie IIB2. Napisz funkcję `najczesciej()`, która dla danego napisu (zakładamy, że występują w nim tylko małe litery alfabetu łacińskiego – tego warunku nie musisz sprawdzać) zwróci pojedynczy znak (`char`), który w napisie wejściowym występuje najczęściej (np. dla "abaacab" będzie to 'a') – w przypadku niejednoznaczności zwróć dowolną wartość spełniającą tę własność.

Kod rozwiązania wraz z komentarzami:

Zadanie IIB3. Napisz funkcję `Loteryjka()`, która zasymuluje losowania pewnej loterii. Przyjmuje ona jako argument tablicę 5 różnych liczb całkowitych z zakresu od 1 do 100 (ułożonych w dowolnej kolejności; faktu, że liczby są różne, nie musisz sprawdzać), reprezentującą liczby typowane przez gracza (czyli jego/jej kupon) oraz dodatnią liczbę całkowitą k . Funkcja ta ma symulować k losowań wspomnianej wyżej loterii – w każdej z k iteracji generowanych jest 5 *różnych* liczb całkowitych z zakresu od 1 do 100. Jako wynik zwracana jest k -elementowa tablica liczb całkowitych, której elementy zawierają informację, jaką kwotę wygrywa gracz w kolejnych odsłonach loterii: za zgadnięcie jednej liczby wygrywa 100 zł. Dla przykładu, założmy, że gracz typuje 1, 6, 9, 43, 29 oraz że decyduje się grać dwukrotnie. W pierwszym losowaniu wygenerowane zostały liczby 1, 2, 3, 4, 6, a w drugim – 1, 8, 29, 43, 44. Oczekiwanym wynikiem powinna być zatem tablica (200, 300).

Wskazówka: Przydatna może być funkcja `rand()`, która generuje liczby całkowite z przedziału od 0 do `RAND_MAX-1` włącznie. Uważaj na powtórzenia liczb przy losowaniu (przykładowo, możesz generować liczby dopóki nie znajdziesz takiej, która się jeszcze nie pojawiła).

Kod rozwiązania i komentarze:

Zadanie IIB4. Wyobraźmy sobie, że żyjemy w krainie, w której każdy rok może mieć inną liczbę miesięcy, a każdy z miesięcy może mieć inną liczbę dni.

- a) Zadeklaruj strukturę `Rok`, która zawiera informację o tym, ile jest miesięcy w danym roku (liczba całkowita) i ile dni ma każdy z tych miesięcy (tablica liczb całkowitych).
- b) Napisz funkcję `stworzRok()` (przyjmującą dwa argumenty: liczbę miesięcy i liczbę dni), która stworzy strukturę `Rok` odpowiadającą rokowi o podanej liczbie miesięcy, w którym każdy miesiąc ma tyle samo dni (szczególny przypadek naszego „mikroświata”).
- c) Napisz funkcję `czyPoprawna` (przyjmującą obiekty typu `Rok` – niekoniecznie wygenerowany przez `stworzRok()` – oraz aktualną datę, tzn. numer dnia i miesiąca – dwie liczby całkowite), która sprawdzi, czy podana data jest poprawna w podanym roku.

Przykład: data „czterdziesty dzień czwartego miesiąca” jest niepoprawna w roku, w którym drugi miesiąc ma mniej niż czterdzieści dni lub w którym są mniej niż cztery miesiące. Uwaga: dni i miesiące numeruj zaczynając od 1.

Kod rozwiązania i komentarze: