

Nazwisko i imię: _____ Nr indeksu: _____

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

| 1 | 2 | 3 | 4 | Σ |
|---|---|---|---|----------|
| | | | | (20) |

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 rozwiązywać. Powodzenia!

Zadanie IA1. Dana jest funkcja zadeklarowana jako:

```
bool czyPierwsza(int n);
```

która orzeka, czy dana liczba jest pierwsza, oraz funkcja

```
int fun(int n);
```

o której nic nie wiemy (bo nie musimy).

Napisz funkcję `g()`, która jako argument przyjmuje liczbę dodatnią `k` i zwraca odpowiedź na pytanie: „Czy dla `k`-tej liczby pierwszej wartość funkcji `fun()` jest liczbą pierwszą?”

Przykład: Przypuśćmy, że wartości funkcji `fun()` są następujące:

| | | | | | | | | | | | | |
|-----------------------------|---|---|---|----|----|---|----|---|---|----|----|-----|
| <code>n</code> | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | ... |
| <code>fun(n)</code> | 1 | 3 | 7 | -8 | -4 | 9 | -6 | 2 | 4 | 7 | 16 | ... |
| <code>czyPierwsza(n)</code> | f | t | t | f | t | f | t | f | f | f | t | ... |

Wówczas `g(2)` zwróci `true` (bo dla liczby 3 – która jest drugą liczbą pierwszą – funkcja `fun()` ma wartość 7), natomiast dla `g(5)` zwróci `false` (bo 16 nie jest liczbą pierwszą).

Kod rozwiązania wraz z komentarzami:

Zadanie IA2. Wojna to jedna z najprzyjemniejszych i najprostszych gier karcianych. Jednak nawet najprostsza gra może sprawiać niektórym trudności. Pomóż Tomkowi i Frankowi zagrać w wojnę pisząc funkcję `wojna()`, która przyjmuje jako argumenty dwa wskaźniki do tablic liczb całkowitych o długości dokładnie 26 elementów, które reprezentują karty. Funkcja powinna zwracać wartość całkowitą:

- a) 1 – jeśli wygrał pierwszy gracz,
- b) 0 – był remis,
- c) -1 – gdy wygrał gracz drugi.

Tablice przekazywane do funkcji reprezentują karty na rękach graczy. i -ty element tablicy to i -ta w kolejności karta na stosach obu graczy. Zawodnicy kolejno zrzucają po jednej karcie i porównują ich wartości, wyższa wartość to punkt dla gracza (zakładamy, że $2 < 3 < \dots < 10 < W < D < K < A$ oraz że kolory kart nie grają roli). W przypadku remisu (takich samych wartości kart) nikt nie dostaje punktu (oryginalna wersja gry w wojnę jest nieco bardziej skomplikowana – tutaj znacząco upraszczamy zasady). Ostatecznie grę wygrywa osoba z większą liczbą punktów.

Karty do gry są reprezentowane przez liczby całkowite od 0 do 51:

| | | | | | | | | | |
|-----------|----|----|----|----|----|----|-----|----|----|
| Kod karty | 0 | 1 | 2 | 3 | 4 | 5 | ... | 50 | 51 |
| Znaczenie | 2♣ | 2♦ | 2♥ | 2♠ | 3♣ | 3♦ | ... | A♥ | A♠ |

Poniżej przedstawiono przykładowe tablice dla gracza pierwszego oraz drugiego. Gracze odkrywają swoje pierwsze karty. Gracz I ma 2♦, gracz II ma 2♥, więc jest remis – nikt nie dostaje punktów. Następnie wykładają 2♣ oraz 5♦ – punkt dla gracza II. W następnym ruchu wykładają odpowiednio 5♥ oraz 4♣ – tym razem punkt dostaje gracz I itd.

| | | | | |
|----------|---|----|----|-----|
| gracz I | 1 | 0 | 14 | ... |
| gracz II | 2 | 13 | 8 | ... |

Kod rozwiązania wraz z komentarzami:

Nazwisko i imię: _____ Nr indeksu: _____

Zadanie IA3. Wszyscy wiemy, jak trudno jest dodawać miesiące z tzw. zawijaniem. Np. ktoś byłby w stanie powiedzieć, jaki to będzie miesiąc, który następuje 17 miesięcy po listopadzie?

Napisz funkcję `dodajMiesiace()`, która będzie rozwiązywała ten dokuczliwy problem. Funkcja przyjmuje jako argumenty: tablicę miesięcy (do których będziemy dodawać, w przykładzie powyżej były to listopad; miesiące numerujemy od 1, czyli styczeń to 1, a grudzień to 12), tablicę miesięcy, które dodajemy (w przykładzie powyżej byłoby to 17, ponieważ chcemy dodać 17 miesięcy) oraz oczywiście rozmiary tych tablic.

Zakładamy, że długość pierwszej tablicy jest dłuższa bądź równa długości drugiej tablicy. W przypadku, gdy druga tablica jest krótsza, trzeba ją „zawijać”.

Przykład (pseudokod):

```
miesiace = (1,2,3,11,12) //styczeń, luty, marzec, listopad, grudzień
dodaj = (1,10,10,4,5)
dodajMiesiace(miesiace,5,dodaj,5) == (2,12,1,3,5) //luty, grudzień, styczeń, marzec, maj

miesiace = (1,2,3,11,12) //styczeń, luty, marzec, listopad, grudzień
dodaj = (1,2,3) // zawijamy do długości 5, czyli jak (1,2,3,1,2)
dodajMiesiace(miesiace,5,dodaj,3) == (2,4,6,12,2) //luty, kwiecień, czerwiec, grudzień, luty
```

Kod rozwiązania i komentarze:

Zadanie IA4. Napisz funkcję o nazwie `przesunL()`, która jako argumenty przyjmuje: tablicę liczb całkowitych x , jej rozmiar n oraz nieujemną liczbę całkowitą k . Zakładamy, że użytkownik podaje zawsze $n, k > 0$ (tego warunku nie musisz sprawdzać).

Funkcja ma zwracać nową, dynamicznie alokowaną tablicę, która powstaje przez “przesunięcie” wszystkich elementów z x o k pozycji w lewo (z zawijaniem).

Dla przykładu, jeśli $x == (1, 2, 3, 4, 5)$ oraz $k == 2$, to w wyniku powinniśmy otrzymać tablicę $(3, 4, 5, 1, 2)$.

Kod rozwiązania i komentarze:

Nazwisko i imię: _____ Nr indeksu: _____

(brudnopis)

(brudnopis)