

## KARTKÓWKA 1

### Zadanie 1

W urnach numerowanych od 1 do  $2n+1$  znajdują się  $n+1$  żetonów czerwonych i  $n$  żetonów białych ułożonych we flagę polską - najpierw czerwone, a potem białe.

Mając do dyspozycji funkcję  $Z(i, j)$  zamieniającą miejscami urnę  $i$ -tą z  $j$ -tą doprowadź do tego, aby żadne dwa żetony tego samego koloru nie sąsiadowały ze sobą. Napisz stosowny algorytm oraz podaj niezmiennik głównej pętli.

Uwaga: robot wykonujący zamiany może się zepsuć, jeśli którykolwiek z argumentów funkcji  $Z$  będzie spoza zakresu  $\{1, \dots, 2n+1\}$  lub jeśli każemy mu zamienić urny o tym samym numerze. Załóż, że  $n \geq 0$ .

### Zadanie 1

*W urnach numerowanych od 1 do  $2n+1$  znajdują się  $n+1$  żetonów czerwonych i  $n$  żetonów białych ułożonych we flagę polską - najpierw czerwone, a potem białe. Mając do dyspozycji funkcję  $Z(i, j)$  zamieniającą miejscami urnę  $i$ -tą z  $j$ -tą doprowadź do tego, aby żadne dwa żetony tego samego koloru nie sąsiadowały ze sobą. Napisz stosowny algorytm oraz podaj niezmiennik głównej pętli.*

*Uwaga: robot wykonujący zamiany może się zepsuć, jeśli którykolwiek z argumentów funkcji  $Z$  będzie spoza zakresu  $\{1, \dots, 2n+1\}$  lub jeśli każemy mu zamienić urny o tym samym numerze. Załóż, że  $n \geq 0$ .*

### Zadanie 1; (20')

15-10-2014

W urnach numerowanych od 1 do  $n$  znajdują się żetony czerwone lub białe. Masz do dyspozycji funkcję  $Kol(i)$  przyjmującą wartość ze zbioru  $\{\text{czerwone}, \text{białe}\}$ . Nadaj całkowitoliczbowej zmiennej  $JedenSasiad$  wartość 1, jeśli żetony w urnach są ułożone tak, że każda urna ma dokładnie jednego sąsiada z żetonem innego koloru, a wartość 0, jeśli tak nie jest. Postaraj się wykonać jak najmniej wywołań funkcji  $Kol$ . Napisz stosowny algorytm w postaci fragmentu programu. Określ, dla jakich danych program wykona najwięcej wywołań funkcji  $Kol$  oraz podaj dokładną liczbę tych wywołań.

Uwaga: robot wykonujący funkcję  $Kol$  może się zepsuć, jeśli jej argument będzie spoza zakresu  $\{1..n\}$ . Załóż, że  $n \geq 0$  (w przypadku  $n=0$  przyjmij, że nie ma żadnej urny).

### Zadanie 1

W urnach numerowanych od 1 do  $4n$  znajdują się żetony czerwone i białe na przemian, tak że na wszystkich pozycjach nieparzystych są żetony czerwone, a na parzystych białe. Mając do dyspozycji procedurę  $Z(i, j)$  zamieniającą miejscami urnę  $i$ -tą z  $j$ -tą doprowadź do tego, aby żetony ułożyły się we „flagę austriacką”: najpierw  $n$  żetonów czerwonych, potem  $2n$  białych i potem znowu  $n$  czerwonych. Napisz stosowny algorytm oraz podaj niezmienniki pętli. Postaraj się wykonać jak najmniej zamian i podaj ich liczbę.

Uwaga: robot wykonujący zamiany może się zepsuć, jeśli którykolwiek z argumentów funkcji  $Z$  będzie spoza zakresu  $\{1, \dots, 4n\}$  lub jeśli każemy mu zamienić urny o tym samym numerze. Załóż, że  $n > 0$ .

Zadanie 1; (20')

17-10-2012

W urnach numerowanych od 1 do  $n$  znajdują się żetony czerwone lub białe. Masz do dyspozycji funkcję  $Kol(i)$  przyjmującą wartość ze zbioru {czerwone, białe}. Nadaj całkowitoliczbowej zmiennej  $FlagaPolska$  wartość 1, jeśli żetony w urnach są ułożone tak, że wszystkie czerwone żetony są w urnach o indeksach mniejszych niż wszystkie białe, a wartość 0, jeśli tak nie jest. Postaraj się wykonać jak najmniej wywołań funkcji  $Kol$ . Napisz stosowny algorytm w postaci fragmentu programu. Określ, dla jakich danych program wykona najwięcej wywołań funkcji  $Kol$  oraz podaj dokładną liczbę tych wywołań.

Uwaga: robot wykonujący funkcję  $Kol$  może się zepsuć, jeśli jej argument będzie spoza zakresu  $\{1..n\}$ . Załóż, że  $n \geq 0$  (w przypadku  $n=0$  przyjmij, że nie ma żadnej urny).

## KARTKÓWKA 2

### Zadanie 2 (20')

22-10-2014

W tablicy  $A[1..n]$  znajdują się tylko zera lub jedynki. Napisz fragment programu, który zmiennej `ile01` nada wartość  $|\{(i,j) : i < j, A[i]=0, A[j]=1\}|$ .

### Zadanie 2 (20')

28-10-2015

W tablicy `int A[n]` znajdują się tylko zera lub jedynki. Napisz fragment programu, który zmiennej `ile` nada wartość  $|\{(i,j) : i < j, A[i] \leq A[j]\}|$ . Określ, jaka jest złożoność algorytmu.

### Zadanie 2

Napisz funkcję `ilezer(n:Integer):Integer` (lub fragment programu nadający zmiennej `ilezer` stosowną wartość), która obliczy iloma zerami kończy się liczba  $n!$ . Możesz korzystać z funkcji `div` i `mod`, które wyznaczają odpowiednio wynik i resztę z dzielenia. Uzasadnij rozwiązanie. Podpisz kartkę z rozwiązaniem podając inicjały prowadzącego ćwiczenia.

### Zadanie 2

Dana jest deklaracja:

**`var t : array [1 .. 3] of Integer;`**

Napisz fragment programu wypisujący na wyjście rozmiar zbioru (bez powtórzeń) elementów tablicy `t`.

Podaj i postaraj się zminimalizować liczbę porównań elementów tablicy `t`, którą Twój algorytm wykona w przypadku pesymistycznym.

Założ, że koszt każdego z porównań „`a=b`”, „`a<>b`”, „`a<b`”, „`a<=b`”, „`a>b`” i „`a>=b`” jest taki sam.

### Zadanie 2

Napisz funkcję `ilezer(n:Integer):Integer` (lub fragment programu nadający zmiennej `ilezer` stosowną wartość), która obliczy iloma zerami kończy się liczba  $n!$ . Możesz korzystać z funkcji `div` i `mod`, które wyznaczają odpowiednio wynik i resztę z dzielenia. Uzasadnij rozwiązanie. Podpisz kartkę z rozwiązaniem podając inicjały prowadzącego ćwiczenia.

### Zadanie 2; (20')

24-10-2012

Napisz funkcję `ile_trójek(n:Integer):Integer` (lub fragment programu nadający zmiennej `ile_trójek` stosowną wartość), która obliczy, jaka potęga trójki występuje w rozkładzie liczby  $n!$  na czynniki pierwsze. Możesz korzystać z funkcji `div` i `mod`, które wyznaczają odpowiednio wynik i resztę z dzielenia. Uzasadnij rozwiązanie. Podpisz kartkę z rozwiązaniem podając poza swoim imieniem i nazwiskiem także inicjały prowadzącego ćwiczenia.



## KARTKÓWKA 3

### Zadanie 3

W urnach numerowanych od 1 do  $n$  (dla  $n > 0$ ) znajdują się żetony białe, czerwone lub niebieskie. Napisz funkcję `ZaniebieszczoneTrójki(n: Integer): Boolean`, która przyjmie wartość `true` wtedy i tylko wtedy gdy w każdej trójce kolejnych (sąsiednich) urn znajduje się choć jeden niebieski żeton. Masz do dyspozycji funkcję `Kol(i)`, która przyjmuje jedną z wartości ze zbioru {biały, czerwony, niebieski}, ale której wywołanie jest drogie. Napisz stosowny algorytm oraz podaj niezmiennik pętli. Podaj też liczbę wywołań funkcji `Kol`. Pamiętaj o możliwej awarii robota, gdy wywołanie funkcji `Kol` będzie spoza zakresu  $[1..n]$ .

### Zadanie 3

O tablicy powiemy że jest  $k$ -płaska, jeśli wszystkie kolejne  $k$ -elementowe segmenty mają równe wartości. Na przykład 3-płaskie są tablice zawierające (jednocyfrowe) wartości 222, 444, 111, natomiast nie są nimi 2221, 123123, ani 44. Dana jest tablica `A: array[0..n] of Integer`, gdzie  $n \geq 0$ , oraz liczba  $k$ . Napisz fragment programu nadający zmiennej  $k$ -płaska wartość `true` jeśli tablica `A` jest  $k$ -płaska, a `false` w przeciwnym razie.

### Zadanie 3

Napisz fragment programu, który dla tablicy `A: array[1..n] of Integer` oraz stałej  $m$  takiej, że  $1 \leq m \leq n$  znajdzie segment długości  $m$  o minimalnej sumie i wartość tej sumy wypisze na standardowe wyjście. Podaj dokładną liczbę operacji arytmetycznych, które Twój algorytm wykona.

#### Wstęp do programowania – podejście imperatywne Kartkówka 3

W tablicy `T: array[1..N] of integer` zapisane są kwoty transakcji z jednego roku, zaś w tablicy `M: array[1..N] of integer` miesiące w których transakcje pod odpowiadającymi indeksami miały miejsce – transakcja o kwocie `T[i]` miała miejsce w miesiącu `M[i]`.

Zakładając, że  $N > 0$  i `M` jest posortowana niemalejąco, proszę napisać fragment programu, który w tablicy `A: array[1..12] of integer` umieści sumaryczne kwoty transakcji dla każdego miesiąca. Sumaryczne kwoty transakcji w żadnym miesiącu nie mogą przekroczyć 1000 jednostek. Jeżeli sytuacja taka ma miejsce, należy nadwyżkę dodać do transakcji w następnym miesiącu. Nadwyżkę powstałą po ostatnim miesiącu należy umieścić w zmiennej `nad: integer`.

Na przykład dla tablic:

`T = 500, 800, 900, 100, 2800`

`M = 3, 3, 4, 8, 11`

wynikiem powinno być

`A = 0, 0, 1000, 1000, 200, 0, 0, 100, 0, 0, 1000, 1000`

`nad = 800`

### Zadanie 3

Dana jest tablica `Int A[n]`, gdzie  $n \geq 0$ . Napisz fragment programu nadający zmiennej `bool HopHop` wartość `true` jeśli tablica `A` zawiera 3-elementowy segment rosnący, a `false` w przeciwnym razie. Segment, to spójny fragment tablicy. Postaraj się zminimalizować liczbę odwołań do elementów tablicy i określ ją dla przypadku pesymistycznego.

4.11.2015

## KARTKÓWKA 4

### Zadanie 4

Ciąg  $A[1], \dots, A[n]$  nazwiemy słabo v-bitonicznym, jeśli istnieje takie  $j$ , że  $1 \leq j \leq n$  oraz dla każdego  $k$ :  $(1 \leq k < j \Rightarrow A[k] \geq A[k+1])$  oraz  $j \leq k < n \Rightarrow A[k] \leq A[k+1]$ .

Napisz fragment programu, który wypisze słowo „NIE“, jeśli tablica nie jest słabo v-bitoniczna lub poda indeks najmniejszego  $j$  spełniającego powyższy warunek v-bitoniczności, jeśli jest słabo v-bitoniczna.

Podaj niezmienniki użytych pętli.

### Zadanie 4

Podaj gramatykę generującą poprawne wyrażenia nawiasowe, ale takie, że nie ma żadnych „samotnych” nawiasów otwierających, tzn. obok każdego nawiasu otwierającego jest co najmniej jeden inny nawias otwierający. Na przykład, słowem poprawnym jest  $((((( ))) ) )$ , a niepoprawnymi są:  $((, (( )), ))(($  (bo to nie są poprawne wyrażenia nawiasowe),  $( )(( ), ((( ) ) )$  (bo podkreślony nawias nie ma sąsiada).

Krótko uzasadnij rozwiązanie, opisując słownie znaczenie każdego symbolu pomocniczego. Podpisz kartkę z rozwiązaniem podając inicjały prowadzącego ćwiczenia.

## Wstęp do programowania – podejście Imperatywne

### Kartkówka 4

Niech  $A : \text{array}[1..N] \text{ of Integer}$ ,  $k : \text{Integer}$ ,  $N > 0$ ,  $k > 0$ . Plateau w tablicy  $A$  to jej spójny fragment  $A[i..j]$ ,  $1 \leq i \leq j \leq N$ , taki że dla każdego  $1 \leq k \leq j$ ,  $A[k] = A[i]$ . Długością plateau jest w tym przypadku  $j - i + 1$ . Proszę napisać fragment programu, który zmiennej  $m : \text{Integer}$  nada wartość równą liczbie plateau w tablicy  $A$  o długości większej niż  $k$ . Na przykład dla  $A = [1, 1, 0, 0, 0, 2, 3, 4]$  i  $k = 1$ ,  $m$  powinno przyjąć wartość 3. Wyznacz największą liczbę operacji dostępu do tablicy, którą Twój algorytm wykona. Dla jakich danych to się stanie?

### Zadanie 4

Dane są dwie tablice  $A$  i  $B$  typu  $\text{tab} = \text{array}[1..n] \text{ of Integer}$ , obie zawierające liczby naturalne (tj. całkowite dodatnie) i posortowane niemalejąco. Napisz funkcję wspólnie(const A,B:tab):Integer, która olicza liczbę różnych liczb naturalnych  $k$  takich, że  $k$  występuje zarówno w  $A$ , jak i w  $B$ . Twoja funkcja nie może korzystać z innych tablic niż  $A, B$ . Podaj koszt pesymistyczny wykonania funkcji w zależności od  $n$ .  
Przykład: Dla tablic zawierających ciągi 1, 1, 1, 2, 4 oraz 1, 1, 3, 4, 4 wynikiem powinno być 2.

### Zadanie 4

Napisz gramatykę, która generuje wszystkie słowa  $v$  nad alfabetem  $\{a,b\}$  takie, że moduł różnicy liczby liter  $a$  w słowie  $v$  i liczby liter  $b$  w słowie  $v$  nie przekracza 2.

18.11.2015

## Wstęp do programowania - podejście Imperatywne

### Kartkówka 4

Dany jest typ  $\text{Tab} = \text{array}[1..N*M] \text{ of Integer}$  reprezentujący macierz  $N$  wierszy na  $M$  kolumn zapisaną wiersz za wierszem. Oznacza to, że w tablicy typu  $\text{Tab}$  pierwszy wiersz mieści się w pierwszych  $M$  komórkach, drugi w kolejnych  $M$  komórkach, itd. Napisz funkcję posortowana(const A:Tablica): Boolean, która przyjmie wartość true wtedy i tylko wtedy gdy każdy wiersz jest posortowany niemalejąco. Postaraj się użyć jak najmniej operacji arytmetycznych, przede wszystkim mnożeń i dodawań. Oblicz, ile dokładnie operacji mnożenia a ile dodawania wykona Twój program w pesymistycznym przypadku. Używanie funkcji break, exit itp. będzie uznawane za usterkę.

Jeśli wciąż masz kłopoty z funkcjami i procedurami, to kosztem 1 punktu możesz napisać fragment programu, który dla zadanej macierzy  $A$  nada zmiennej posortowana stosowną wartość. Załóż, że  $M \geq 1$  i  $N \geq 1$ .

07/11/12

## KARTKÓWKA 5

Kartkówka nr 5.

17.11.2011

Dany jest typ `tab = array [1..n] of Integer` i dwie tablice A i B tego typu. Tablica A jest posortowana niemalejąco, a tablica B nierosnąco i obie zawierają wyłącznie zera lub jedynki. Napisz funkcję `tasama(const A,B:tab):Boolean`, która przyjmie wartość `true` wtedy i tylko wtedy gdy istnieje taki indeks  $i$ , że  $1 \leq i \leq n$  oraz  $A[i] = B[i]$ . Odpowiedź uzasadnij i podaj koszt pesymistyczny wykonania funkcji w zależności od  $n$ .

### Zadanie 5

Napisz gramatykę generującą język takich słów nad alfabetem  $\{0, 1, 2\}$ , w których pomiędzy każdymi dwoma zerami występuje jedynka, pomiędzy każdymi dwoma jedynkami występuje zero, (dwójki mogą występować dowolnie), np: 1, 1010, 0221021. Jeśli oprócz symbolu startowego użyjesz innych symboli pomocniczych, opisz krótko język generowany przez każdy symbol.

### Zadanie nr 5.

Dany jest typ `tab = array [1..n] of Integer` i dwie tablice A i B tego typu.

Tablica A jest posortowana niemalejąco, a tablica B nierosnąco i obie zawierają wyłącznie zera lub jedynki. Napisz funkcję `tasama(const A,B:tab):Boolean`, która przyjmie wartość `true` wtedy i tylko wtedy gdy istnieje taki indeks  $i$ , że  $1 \leq i \leq n$  oraz  $A[i] = B[i]$ . Odpowiedź uzasadnij i podaj koszt pesymistyczny wykonania funkcji w zależności od  $n$ .

### Wstęp do programowania – kartkówka nr 5.

21 listopada 2012

Dana jest tablica A typu `array [1..N] of integer`, wypełniona zerami i jedynkami. Zakładając, że  $N > 1$ , przypisz zmiennej `naleszy` wartość `true` wtedy i tylko wtedy gdy słowo zapisane w tablicy A należy do  $L(G)$ , gdzie  $G = (\{X, Y\}, \{0, 1\}, P, X)$  i  $P$  to następujący zbiór produkcji:

$X \rightarrow 0X1 \mid 0Y1 \mid 0 \mid 1 \mid \epsilon$

$Y \rightarrow 1Y0 \mid 1X0 \mid 0 \mid 1 \mid \epsilon$ .

### Zadanie 5 (20')

Dana jest tablica `int A[n]` posortowana niemalejąco. Napisz fragment programu, który zmiennej `MinModul` nada wartość równą modułowi liczby o najmniejszym module. Określ złożoność algorytmu.

25-11-2015

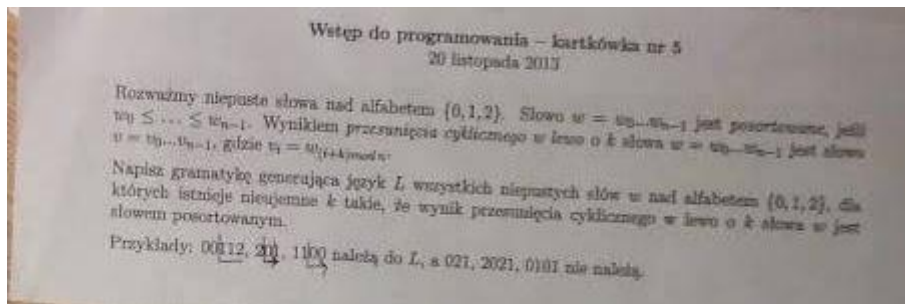


Wstęp do programowania – kartkówka nr 5  
12 listopada 2014

Niech alfabet  $A$  składa się z liter  $a, b, c$ . Kostką domina nazywamy dowolne dwuliterowe słowo nad alfabetem  $A$ . Kostka  $xy$  pasuje do kostki  $yz$  dla dowolnych  $x, y, z \in A$ . Poprawną rozgrywką domina nazywamy dowolne słowo nad alfabetem  $A$ , takie że kolejne dwuliterowe podsłowa tego słowa tworzą pasujące do siebie kostki domina. Na przykład,  $\epsilon, aa, abbcab$  są poprawnymi rozgrywkami, natomiast  $abab, aaa$  nie są.

Napisz gramatykę generującą poprawne rozgrywki domina nad  $A$ .

wszystkie



## KARTKÓWKA 6

Kartkówka nr 6.

30.11.2011

W problemie wieży Hanoi mamy dodatkowe ograniczenie mobilności krążków. Krążki o nieparzystych numerach można przenosić zgodnie ze wskazówkami zegara:  $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$ , zaś krążki o parzystych numerach - przeciwnie:  $1 \rightarrow 3 \rightarrow 2 \rightarrow 1$ . Pozostałe reguły przenosin są bez zmian. Napisz procedurę  $H(n,s,d:\text{Integer})$ , która wykona przenosiny  $n$  krążków z wieży o numerze  $s$  na wieżę o numerze  $d$ .

### Zadanie 6

Napisz funkcję

**RóżneReszty (const A: Array [1 .. n] of Integer): Boolean**

przyjmującą wartość true wtedy i tylko wtedy, gdy w każdym trzelementowym segmencie tablicy A wszystkie reszty z dzielenia znajdujących się tam wartości przez 3 będą różne. Postaraj się zminimalizować liczbę wywołań funkcji mod. Nie chodzi o to, aby w ogóle jej nie wywoływać, tylko żeby nie wywoływać niepotrzebnie, na przykład wielokrotnie dla tych samych argumentów. Zakładamy jedynie, że  $n > 0$ .

Wstęp do programowania – podejście imperatywne  
Kartkówka 6

Proszę napisać fragment programu, który dla tablicy `int a[n]`, takiej, że dla każdego  $2 \leq j \leq n-1$ ,  $a[j-2] > a[j]$  (jeżeli  $j$  parzyste) i  $a[j-2] < a[j]$  (jeżeli  $j$  nieparzyste), nada zmiennej `bool` jest wartość true jeżeli istnieje  $0 \leq i \leq n-2$  takie, że  $a[i] = a[i+1]$ , oraz wartość false w przeciwnym przypadku.

Wstęp do programowania - kartkówka nr 6 (27 listopada 2013)

Dane są deklaracje:

```
const N = ...; {N >= 1}
```

```
type Tablica = array [1 .. N] of Integer;
```

Napisz funkcję

```
function uporządkowaneModuly(const t : Tablica) : Boolean;
```

która sprawdzi, dla liczb zapisanych w porządku niemalejącym w tablicy `t`, czy ciąg ich wartości bezwzględnych jest również uporządkowany niemalejąco.

Wstęp do programowania – podejście imperatywne  
Kartkówka 6

Dana jest tablica `T : array [1..N] of integer`,  $N > 0$ , w której różnice między kolejnymi wartościami, są posortowane ściśle rosnąco, tzn. dla każdego  $i \in \{1, \dots, N-2\}$ ,  $T[i+1] - T[i] < T[i+2] - T[i+1]$ . Proszę napisać fragment programu, który zmiennej `istnieje : boolean` przypisze wartość true wtedy i tylko wtedy gdy w `T` istnieją dwie sąsiadujące komórki o takiej samej wartości.



## KARTKÓWKA 7

Kartkówka nr 7.

7.12.2011

W dwóch tablicach  $A, B: \text{array}[0..n]$  of  $0..1$  znajdują się bity rozwinięcia dwójkowego dwóch liczb całkowitych zapisanych w układzie uzupełnieniowym do 2 (czyli  $U_2$ ) dla pewnego  $n \geq 0$ , przy czym bit  $[n]$  odpowiada znakowi, a bity  $[k]$  wartościom  $2^k$  dla  $0 \leq k < n$ .  
Napisz funkcję `mniejsze(const A, B: array[0..n] of 0..1): Boolean`, która przyjmie wartość `true` wtedy i tylko wtedy, gdy liczba reprezentowana przez  $A$  jest mniejsza niż liczba reprezentowana przez  $B$ .

### Zadanie 7

Tablica  $A$  zawiera posortowane rosnąco liczby naturalne reprezentujące pewien  $n$ -elementowy zbiór  $A$ ,  $n > 0$ . Napisz funkcję

**RóżnePodzbiory** (`const A: Array [1 .. n] of Integer; S: Integer`): `Integer`

która wyznaczy liczbę różnych podzbiorów zbioru  $A$  o sumie elementów równej  $S$ .  
Określ złożoność czasową i pamięciową podanego algorytmu.

Zadanie nr 7. 7.12.2011

W dwóch tablicach  $A, B: \text{array}[0..n]$  of  $0..1$  znajdują się bity rozwinięcia dwójkowego dwóch liczb całkowitych zapisanych w układzie uzupełnieniowym do 2 (czyli  $U_2$ ) dla pewnego  $n \geq 0$ , przy czym bit  $[n]$  odpowiada znakowi, a bity  $[k]$  wartościom  $2^k$  dla  $0 \leq k < n$ .

Napisz funkcję `mniejsze(const A, B: array[0..n] of 0..1): Boolean`, która przyjmie wartość `true` wtedy i tylko wtedy, gdy liczba reprezentowana przez  $A$  jest mniejsza niż liczba reprezentowana przez  $B$ .

Kartkówka nr 7

5 grudnia 2012

Rozważamy uogólnienie problemu wieży Hanoi. Reguły przenoszenia pozostają bez zmian, ale wieżę docelową wskazujemy niezależnie dla każdego krążka.

Wartości typu `Tablica = array [1 .. N] of 1 .. 3` określają wieże docelowe dla  $N$  krążków, w kolejności od krążka najmniejszego. Procedura `jeden(skad, dokad : Integer)` przenosi jeden krążek z wieży nr `skad` na wieżę nr `dokad`.

Zdefiniuj procedurę `uporządkuj(const t : Tablica)`, która zgodnie z regułami przeniesie  $N$  krążków z wieży pierwszej na wieże określone tablicą  $t$ . Podaj liczbę wywołań procedury `jeden` w przypadku pesymistycznym.

Zadanie 7

09.12.2015

Krawędzie skierowanego grafu o trzech wierzchołkach numerowanych liczbami  $0, 1, 2$  reprezentowane są w tablicy `bool G[3][3]`. W tej tablicy `G[i][j] = true` dla  $0 \leq i, j \leq 2$  wtedy i tylko wtedy gdy istnieje krawędź między wierzchołkiem  $i$  oraz wierzchołkiem  $j$ . Graf jest silnie spójny, jeśli dla każdej pary wierzchołków  $(i, j)$  istnieje niepusty ciąg krawędzi prowadzących od wierzchołka  $i$  do wierzchołka  $j$ . Napisz funkcję `bool SilnieSpójny(bool *G)`, która przyjmie wartość `true`, gdy graf reprezentowany w tablicy  $G$  jest silnie spójny, a `false` gdy nie jest.

Napisz funkcję `merge`, która scala dwie posortowane niemalejąco tablice `int A[N]` oraz `int B[N]` do posortowanej niemalejąco tablicy `int C[2*N]` bez używania konstrukcji pętli (`for/while/do`). Oprócz  $A$ ,  $B$  i  $C$ , funkcja `merge` może przyjmować również inne parametry; wyjaśnij ich znaczenie.  $N$  jest pewną zdefiniowaną stałą większą od 0.

### Wstęp do programowania - kartkówka nr 7 (4 grudnia 2013)

Rozważamy problem wieży Hanoi z parzystą liczbą krążków. Krążki dzielimy na dwie równoliczne grupy: krążki małe i duże. Przyjmujemy, że krążki duże można przenosić tylko pomiędzy sąsiadującymi wieżami ( $1 \leftrightarrow 2, 2 \leftrightarrow 3$ ). Zdefiniuj procedurę `hanoi(n : Integer)` wypisującą najkrótszy ciąg operacji, który przeniesie  $n$  krążków z wieży 1 na 3.

## KARTKÓWKA 8

Kartkówka nr 8.

21.12.2011

Dany jest typ `tablica = array[1..n] of Integer` ( $n > 0$ ) oraz typ `lista` zdefiniowany jak na wykładzie. Tablica `A` zawiera liczby całkowite posortowane niemalejąco. Napisz funkcję `tylkonieparzyste(const A:tablica):lista`, która utworzy posortowaną rosnąco listę wszystkich nieparzystych wartości z tablicy `A`.

Kartkówka nr 8

12 grudnia 2012

Dane są definicje typów `Cyfra = 0 .. 1`; `Liczba1 = array [0 .. N1 - 1] of Cyfra`; `Liczba2 = array [0 .. N2 - 1] of Cyfra`; dla  $N1$  i  $N2$  większych lub równych 1. Wartości tych typów reprezentują liczby całkowite, zapisane w kodzie U2 w kolejności od cyfr najmniej znaczących.

Zdefiniuj procedurę `przekształc(const a : Liczba1; var b : Liczba2; var ok : Boolean)` która, jeśli da się to zrobić, w tablicy `b` umieści reprezentację  $N2$ -bitową wartości zapisanej w tablicy `a` w reprezentacji  $N1$ -bitowej. Zmienna `ok` ma przyjąć wartość `true` wtedy i tylko wtedy, gdy się udało, a `false` w przeciwnym przypadku.

Wstęp do programowania - podejście imperatywne  
Kartkówka 8

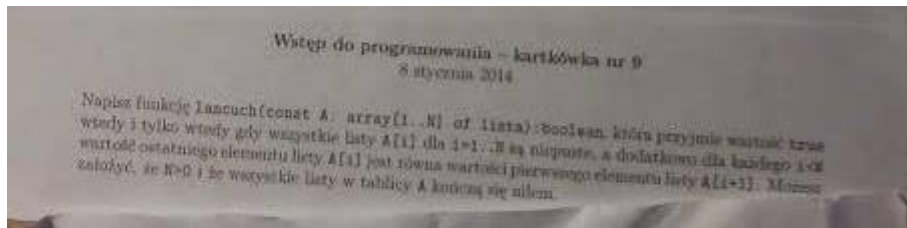
Proszę napisać procedurę `naprzemian(var l1 : lista; l2 : lista)`, która zmodyfikuje listę `l1` tak by zawierała ona na przemian elementy z `l1` i `l2`, zaczynając od elementu z `l1`. Należy założyć, że obie listy są rozłączne, są listami prostymi i mają taką samą liczbę elementów. Na przykład dla list `l1 = 1, 3, 5` i `l2 = 2, 4, 6`, po wykonaniu procedury ma być `l1 = 1, 2, 3, 4, 5, 6`.

### Zadanie 8

Modyfikujemy zasady wieży Hanoi tak, że robimy wyjątek dla najmniejszego krążka o numerze 1 i pozwalamy na niego położyć w każdym ruchu dowolny krążek. Pozostałe reguły przenoszenia krążków są takie, jak w oryginalnym problemie wieży Hanoi. Napisz procedurę `Hanoi1(n:Integer)`, która dokona przełożenia  $n$  krążków z wieży o numerze 1 na wieżę o numerze 3 w postaci ciągu pojedynczych ruchów poprawnych dla tak zmodyfikowanej reguły przenoszenia. Każdy ruch powinien mieć postać wypisania na standardowe wyjście komunikatu „przenieś krążek  $j$  z wieży  $s$  na wieżę  $d$ ” dla pewnych  $j, s, d$  takich, że  $1 \leq j \leq n$ ,  $1 \leq s \leq 3$ ,  $1 \leq d \leq 3$ . Zadbaj o wykonanie możliwie małej liczby kroków; dla  $n > 10$  nie powinno ich być więcej niż  $2^n - 100$ .



## KARTKÓWKA 9



### Zadanie 9

Napisz procedurę

**Rekin (l: lista; var dasie: Boolean)**

która dla danej listy cyklicznej *l*, w której żadne 2 wartości się nie powtarzają, pozamienia tak ze sobą jej elementy, aby żadne trzy kolejne nie tworzyły w powstałej liście cyklicznej ciągu monotonicznego (rosnącego lub malejącego). Jeśli jest to niemożliwe, to parametr **dasie** ma przyjąć wartość false, w przeciwnym razie true. W każdym przypadku kolejność elementów w liście *l* po wykonaniu procedury może różnić się od oryginału, choć lista ma składać się z tych samych elementów i nadal być cykliczna. Uwaga: wolno zamieniać wskaźniki, ale nie wolno przepisywać samych wartości całkowitych między rekordami.

Zadanie nr 9.

4.01.2012

Napisz procedurę *kopia2(l:lista; var l2:lista2)*, tworzącą dwukierunkową listę która zawiera te same wartości, co dana lista *l*. Przykładowo dla listy zawierającej dwie wartości 3 i 7 powinna powstać osobna lista *l2* zawierająca te same wartości, jak na rysunku poniżej.

3

7

3

7

1

12

Wstęp do programowania – kartkówka nr 9  
9 stycznia 2013

*l* -> na jest lista liczb całkowitych 1. Napisz procedurę *uzupełnij(l:lista)*, która między każde dwa kolejne elementy listy o wartościach *n* i *n+2* wstawi element o wartości *n+1*. Na przykład listę 0, 2, 3, 5 zamieni na 0, 1, 2, 3, 4, 5.

Wstęp do programowania – kartkówka nr 9  
9 stycznia 2013

*l* -> na jest lista liczb całkowitych 1. Napisz procedurę *uzupełnij(l:lista)*, która między każde dwa kolejne elementy listy o wartościach *n* i *n+2* wstawi element o wartości *n+1*. Na przykład listę 0, 2, 3, 5 zamieni na 0, 1, 2, 3, 4, 5.



Kartkówka nr 9.

4.01.2012

Napisz procedurę `kopia2(l:lista; var l2:lista2)`, tworzącą dwukierunkową listę która zawiera te same wartości, co dana lista `l`. Przykładowo dla listy zawierającej dwie wartości 3 i 7 powinna powstać osobna lista `l2` zawierająca te same wartości, jak na rysunku poniżej.



### Zadanie 9

Napisz funkcję `mean(l:lista):boolean`, która przyjmie wartość `true` wtedy i tylko wtedy, gdy któryś z elementów listy ma wartość będącą średnią arytmetyczną jego sąsiadów.

## KARTKÓWKA 10

### Kartkówka 10

Napisz funkcję `InnyBrat(d: drzewo): Boolean`, która przyjmie wartość `true` wtedy i tylko wtedy, gdy drzewo `d` jest puste lub dla każdego węzła tego drzewa, który ma brata, wartość pola `w` jest różna od analogicznej wartości dla jego brata.

### Zadanie 10

Głębokością węzła w drzewie nazywamy długość prowadzącej do niego ścieżki od korzenia. Napisz funkcję: **`Dobre(d: drzewo): Boolean`** która sprawdzi, czy w danym jako parametr drzewie binarnym ciąg głębokości liści, w porządku od lewej do prawej, jest (ściśle) malejący.

### Kartkówka nr 10.

Napisz procedurę `NajdluzszyWiersz(s1,s2:string)`, która skopiuje z pliku tekstowego o ścieżce dostępu `s1` tylko najdłuższy wiersz do pliku o ścieżce dostępu `s2`. Postaraj się założyć jak najmniej o długości wierszy pliku `s1`. W przypadku kilku najdłuższych wierszy jednakowej długości skopiuj pierwszy z nich.

18.01.2012

### Zadanie nr 10 18. 01. 2011

W pliku o nazwie `s`, zawierającym liczby rzeczywiste, znajdują się wartości pewnego kopca, który był reprezentowany w tablicy. Dane w pliku odpowiadają kolejnym elementom tablicy. Napisz funkcję `UtworzKopiec(var s:string):drzewo`, która utworzy drzewo binarne reprezentujące kopiec dany w pliku o nazwie `s`.

### Zadanie 10 (20')

23-01-2013

Napisz procedurę `kompresuj(var f:text; var g:file of Byte)`, która z pliku tekstowego złożonego z samych zer i jedynek utworzy plik bajtów, w którym w kolejnych bajtach będą się znajdować liczby naturalne kodujące kolejne ósemki bitów. Przyjmij, że zmienne plikowe występujące w nagłówku zostały już związane z odpowiednimi plikami dyskowymi. Załóż, że łączna liczba zer i jedynek w pliku `f` jest podzielna przez 8 i że nie występują w nim żadne znaki inne niż zera lub jedynki.

### Kartkówka nr 10.

22.01.2014

Dane są dwa pliki tekstowe `f1` i `f2`, otwarte w trybie do odczytu. Każdy z plików w każdym wierszu zawiera ciąg małych liter alfabetu angielskiego. Żaden z plików nie jest pusty ani nie zawiera pustych wierszy. Napisz funkcję `niektore(var f1,f2:Text):Boolean`, której wynikiem jest `true` wtedy i tylko wtedy, gdy plik `f2` można uzyskać z pliku `f1` poprzez usunięcie niektórych wystąpień niektórych liter oraz niektórych całych wierszy. Musisz założyć, że wiersze pliku `f2` zawierają co najwyżej 200 znaków każdy.