Zadania przygotowawcze do I kolokwium ze Wstępu do Programowania w semestrze zimowym 2007/2008 dla kierunku Informatyka

Dane są tablice A,P: array[1..n] of Integer, gdzie n jest stałą większą od 2, przy czym tablica P jest posortowana niemalejąco. Zakładamy, że elementy w obu tablicach są dodatnie.

Postaraj się napisać każde rozwiązanie w formie procedury lub funkcji z wyspecyfikowanymi w nagłówku odpowiednimi parametrami (Na kolokwium procedury nie są wymagane, ale mile widziane).

We wszystkich zadaniach należy umieć uzasadnić rozwiązanie, np. przez podanie niezmiennika głównej pętli algorytmu i oszacować złożoność obliczeniową (czyli ile razy wykonają się podstawowe operacje; przede wszystkim chodzi o rząd wielkości). Większość zadań (ale nie wszystkie!) ma złożoność liniową. Prośba o niezmienniki jest trochę na wyrost – będzie o nich wkrótce na wykładzie już po kolokwium - ale ci z Państwa, którzy już się z nimi zapoznali proszeni są o nie unikanie ich.

1. Zastąp każdy element tablicy A przez dodanie jego wartości do sumy wartości wszystkich elementów go poprzedzających
2. Odtwórz z tablicy otrzymanej w poprzednim zadaniu oryginalne wartości.
3. Zastąp każdy element tablicy A poza pierwszym i ostatnim przez średnią arytmetyczną z sąsiadujących z nim wartości. Załóż, że wszystkie oryginalne wartości w tablicy A są parzyste.
4. Jeżeli to możliwe, odtwórz oryginalne wartości z tablicy uzyskanej w poprzednim zadaniu.
5. Zastąp każdy element tablicy A przez średnią arytmetyczną z sąsiadujących z nim wartości. Załóż, że sąsiadem lewym pierwszego elementu jest element ostatni, a sąsiadem prawym ostatniego --- element pierwszy.
6. Jeżeli to możliwe, odtwórz oryginalne wartości z tablicy uzyskanej w poprzednim zadaniu.
7. Znajdź najmniejszą i największą wartość znajdującą się w A.
8. Znajdź wartość drugiego co do wielkości elementu z A.
9. Znajdź drugą co do wielkości wartość z A. Jeżeli takiej nie ma, to wynikiem niech będzie wartość 0. *Różnica między tym, a poprzednim zadaniem polega na tym, że np. dla tablicy A=(3,1,5,4,5,5) odpowiedzią tutaj powinna być wartość 4, a w poprzednim zadaniu wartość 5.*
10. Zbadaj, ile jest różnych wartości w tablicy P.
11. Przepisz do tablicy A elementy z P tak, aby każda wartość z P pojawiła się w A tylko raz. Elementy tablicy A, do których nie przepisano żadnej wartości, powinny mieć wartość 0.
12. Znajdź wartość, która pojawia się w P najwięcej razy.
13. Zbadaj, czy w tablicy P występują przynajmniej 3 elementy o tej samej wartości.
14. Odbij lustrzanie tablicę A względem jej środka. Przykładowo dla tablicy A wypełnionej wartościami (3,1,5,4,5,5) wynikiem powinna być ta sama tablica A, ale o wartościach (5,5,4,5,1,3).
15. Sprawdź, czy w tablicy A istnieją choć dwa elementy, których wartości są względnie pierwsze.
16. Sprawdź, czy w tablicy A istnieją choć dwa elementy, których wartości nie są względnie pierwsze.
17. Znajdź największy wspólny dzielnik wszystkich wartości występujących w tablicy A.
18. Wyznacz liczbę lokalnych ekstremów w tablicy A. Lokalnym ekstremum nazwiemy taki element A[k], że dla każdego k takiego, że 2 <= k <= n-1, (A[k]-A[k-1])(A[k]-A[k+1])>0.
19. Zakładając, że elementy tablicy A określają długości odcinków. Zbadaj, czy
    a) z każdych 3 odcinków, których długości są reprezentowane w A da się zbudować trójkąt;

b) istnieją takie trzy odcinki, z których da się zbudować trójkąt.

20. Oblicz maksymalna wartość bezwzględną różnicy dwóch różnych elementów tablicy P, przy założeniu, że może ona zawierać liczby ujemne.
21. Sprawdź, czy w dwóch posortowanych niemalejąco tablicach P1, P2 znajdują się choć dwie takie same wartości. Odpowiedź powinna być twierdząca zarówno wtedy, gdy te dwa równe elementy znajdują się w różnych tablicach, jak i wtedy gdy oba są w tej samej.
22. Zaprogramuj podstawowe działania arytmetyczne na liczbach naturalnych wielocyfrowych w układzie pozycyjnym o dowolnej bazie. (+,-,*,div/mod).
23. *Zadanie o fladze austriackiej.* W n urnach znajduje się 2k żetonów czerwonych oraz n-2k białych (n >= 2k >= 0). Dysponując funkcją Z(i,j) zamieniającą zawartość urny i-tej z j-tą oraz funkcją KOL(i) podającą kolor żetonu w i-tej urnie doprowadź do sytuacji, w której wszystkie białe żetony będą znajdowały się pomiędzy dwiema równolicznymi grupami czerwonych.
24. *Zadanie o kwiatkach* Przy stole przygotowano nakrycia dla k pań i n-k panów (0 <= k <= n \ge 1). Nakrycia dla pań były wyróżnione kwiatkiem, który stał przy nakryciu. Zaproszeni goście (k pań i n-k panów) losowo usiedli przy stole nie zauważywszy finezji organizatorów. Dysponując funkcją KWIATEK(i) przyjmującą wartość true wttw gdy przy i-tym nakryciu znajduje się kwiatek, funkcją PANI(i) stwierdzającą, czy przy i-tym miejscu usiadła pani oraz funkcją Z(i,j) zamieniającą osoby siedzące przy i-tym i j-tym nakryciu doprowadź do właściwego usadowienia uczestników bankietu.
25. *Zadanie o paniach* Przy stole usiadły panie i panowie, przy czym panów jest nie mniej, niż pań. Dysponując operacją Z(i,j) oraz funkcją PANI(i) sprawdzającą, czy przy i-tym miejscu siedzi pani, doprowadź do sytuacji, w której żadne dwie panie nie siedzą koło siebie. (Uwaga: minimalizacja liczby zamian jest nietrywialna).
26. *Zadanie o maksymalnej wysokości stoku* W tablicy liczb całkowitych A[1..n] stokiem wznoszącym się (l,p), gdzie 1<= l<= p<= n nazywamy spójny fragment tablicy taki, że A[l] <=...<= A[p]. Wysokość stoku (l,p) wynosi A[p]-A[l]. Napisz fragment programu, który znajdzie maksymalną wysokość stoku w tablicy A.
27. *Zadanie o Pracusiu.* Pracuś postanowił, że będzie pracował tak dużo i często, aby w ciągu kolejnych k dni co najmniej r dni było dniami roboczymi. Napisz fragment programu rozstrzygający, czy Pracuś był faktycznie tak pracowity, jak zamierzał. Kolejne dni kodowane są w tablicy dni:array[1..n] of Boolean, gdzie dni[i]=true oznacza, że Pracuś pracował i-tego dnia, a false, że miał wolne.
28. *Zadanie o różnicy symetrycznej zbiorów* Dane są dwie tablice A[1..n], B[1..n] liczb całkowitych. Liczby w każdej tablicy uporządkowane są (ściśle) rosnąco. Napisz fragment programu, spełniający warunek końcowy r=|A ÷ B|, gdzie operator ÷, to różnica symetryczna. (Warianty: zamiast różnicy symetrycznej rozwiąż zadanie dla sumy, iloczynu, różnicy...)
29. Dane są 3 tablice A,B,C całkowite uporządkowane rosnąco. Napisz fragment programu obliczający liczbę tych elementów, które występują jednocześnie we wszystkich trzech tablicach.
30. W fabryce Bardzo Skomplikowanych Urządzeń zainstalowano taśmę produkcyjną obsługiwaną przez roboty. Każdy robot w danym dniu ma zaprogramowane wykonywanie dokładnie jednej czynności. Czynności oznaczamy liczbami naturalnymi. Aby zmontować urządzenie, należy wykonać określone czynności w zadanym porządku. Dane urządzenie daje się zmontować, jeśli roboty wykonujące wszystkie operacje konieczne dla zmontowania urządzenia są ustawione (niekoniecznie obok siebie) we właściwej kolejności wzdłuż taśmy. W tablicy C: array [1..m] of integer zakodowane są czynności w porządku koniecznym do zmontowania pewnegourządzenia U. W tablicy R:array [1..n] of integer zakodowane są czynności wykonywane przez kolejno stojące przy taśmie roboty. Napisz funkcję logiczną, która przyjmie wartość true wttw gdy urządzenie U daje się wykonać przy taśmie zadanej tablicą R.
31. Dana jest tablica A: array[1..n] of real, n\ge 1. Nadaj zmiennej x wartość równą $\min_{1<= k<= n}$ | ∑1<= i <= k: A[i] -∑ k+1<= i <= n A[i]|
Uzasadnij poprawność programu, najlepiej wpisując odpowiednie niezmienniki i wyjaśniając ich użycie. Podaj maksymalną liczbę porównań i przypisań,

które program wykona.

32. Ciąg liczbowy x1,...,xn jest *bitoniczny*, wtedy i tylko wtedy, gdy istnieje 1 <= m <= n takie, że x1 < x2 < ... < xm > xm+1 > ... > xn.
    a) Napisz funkcję bitoniczny(var A:tab):Boolean, która stwierdzi, czy ciąg zapisany w tablicy A jest bitoniczny.
    b) Napisz funkcję max(var A: tab):Real obliczającą maksimum ciągu bitonicznego umieszczonego w tablicy A.
    c) Napisz procedurę sort(var A, B: tab); przepisującą elementy ciągu bitonicznego umieszczonego w tablicy A do tablicy B w kolejności niemalejącej.
33. W dwóch tablicach T[1..n], S[1..n] (n>0) znajdują się wartości całkowite nieujemne. Napisz fragment programu, który zmiennej logicznej tesamesumy nada wartość true, jeśli istnieją takie indeksy 1<=i,j<=n, że $\sum_{k=1}^{i}T[k]=\sum_{k=1}^{j}S[k]$.
34. Tablica A jest uporządkowana niemalejąco. Przepisz jej wartości do tablicy B tak, aby B była uporządkowana wg. wartości bezwzględnych.
35. Ciąg nazwiemy *naprzemiennym*, jeśli każde dwa sąsiadujące wyrazy ciągu mają przeciwne znaki (przyjmujemy, że zero nie ma znaku). W tablicy T[1..n] of real zadany jest ciąg t_n liczb rzeczywistych. Napisz fragment programu, który obliczy wartość największej sumy elementów pewnego spójnego i naprzemiennego podciągu ciągu t_n, czyli max($\sum_{k=i}^{j}$ t_k: 1 <= i <= j <= n) i ciąg t_i,...,t_j jest naprzemienny
36. Oblicz maksymalną różnice indeksów między elementami o równych wartościach w tablicy A.
37. Sprawdź, czy w tablicy A istnieją takie trzy kolejne elementy, że suma dwóch pierwszych równa jest trzeciemu.
38. Dla n=2k przestaw elementy tablicy A tak, żeby wszystkie wartości A[1..k] były mniejsze lub równe wszystkim wartościom A[k+1..n].
39. Danych jest n liczb zespolonych we współrzędnych biegunowych. Argumenty główne dane są w stopniach w tablicy Arg[1..n] of Real, a moduły w tablicy R[1..n] of Real. Wiadomo, że dla każdego 1<=i<=n: 0<=Arg[i]<=180, R[i]>0. Napisz fragment programu, który zmiennej logicznej kwadrat nada wartość true wtedy i tylko wtedy, gdy istnieje kwadrat o wierzchołku w środku układu współrzędnych zawierający wszystkie dane liczby.
40. Transpozycja permutacji, to zamiana jej dwóch sąsiednich elementów. Dwie permutacje mają tę samą parzystość, jeśli istnieje parzysta liczba transpozycji przekształcających jedną z nich na drugą. Dane są dwie różnowartościowe tablice A,B:array[1..n] of 1..n reprezentujące dwie permutacje (na i-tym miejscu numer elementu, który w wyniku wykonania permutacji tam trafi). Napisz fragment programu, który zmiennej tasama:Boolean nada wartość true wtedy i tylko wtedy, gdy permutacje zapisane w tablicach A i B mają tę samą parzystość.
41. Dana jest tablica C:array[1..m] of Integer o wartościach nieujemnych oraz dwie liczby a,b:Longint. Napisz fragment programu, który zmiennej logicznej jest nada wartość true wtedy i tylko wtedy gdy istnieją takie liczby 1<= i<= j<= m, że a<=$\sum_{k=i}^{j}$ C[k]<= b.
42. Dana jest tablica t: array[1..2*n] of Integer dla stałej całkowitej N większej od zera. Dla i,j takich, że 1<= i<= j<= N mówimy, że spójny fragment tablicy T[i] T[i+1] ... T[j] jest lustrzanym fragmentem w T,jeśli T[i] = T[2*N+1 – i],T[i+1] = T[2*n – i],...,T[j] = T[2*N+1 -j].Napisz fragment programu, który dla danej tablicy T oblicza długość najdłuższego fragmentu lustrzanego w T. Jeżeli T nie ma żadnego fragmentu lustrzanego, to odpowiedź brzmi 0.
43. Powiemy, że tablica T jest posortowana wierszowo wttw gdy każdy wiersz tablicy T jest posortowany, (a kolumnowo gdy każda kolumna tablicy T jest posortowana). Dana jest tablica T[1..n,1..m] of real posortowana zarówno wierszowo jak i kolumnowo oraz liczba x: real. Napisz fragment programu, który ustali wartość zmiennej jest na true wttw gdy w tablicy T istnieje element o wartości x.
44. Napisz fragment programu, który tak poprzestawia elementy tablicy A[1..n,1..m] of real, aby była posortowana zarówno wierszowo, jak i kolumnowo.
45. Tablica A posortowana jest cyklicznie, jeśli istnieje taki indeks 1<=i<=

n, że ciąg A[i+1],A[i+2],...,A[n], A[1],...,A[i] jest posortowany rosnąco. Znajdź w cyklicznie posortowanej tablicy jej największą wartość.

46. Przesuń tablicę A cyklicznie w prawo o 1 pozycję, to znaczy że każdy element poza ostatnim ma się znajdować o jedną pozycję na prawo, a element ostatni znajdzie się na pierwszej pozycji.
47. Przesuń tablicę A cyklicznie w prawo o k pozycji dla dowolnego k całkowitego.
48. Podaj gramatykę generującą język $\{a^kb^ma^n$: k,m,n są nieujemne i takie że $k+m \neq n\}$. Uzasadnij poprawność podanej gramatyki.
49. Podaj gramatykę, która generowałaby wyrażenia arytmetyczne dopuszczające potęgowanie (oznaczone symbolem ^), będące działaniem o najwyższym priorytecie i prawostronnie łącznym (czyli np. wyrażenie 3*2^1^2 + 5*2 powinno dać wartość 16).
50. Podaj gramatykę generującą wszystkie liczby podzielne przez 3
   a) w systemie dziesiętnym,
   b) w systemie dwójkowym.
51. Podaj gramatykę generującą język wszystkich słów nad alfabetem {a,b} takich, że nigdzie nie występują trzy a obok siebie.
52. Podaj gramatykę, która generuje nad alfabetem {(,)} język słów postaci $(^n)^n$ dla n>=0,
   a) dowolnej poza słowami z poprzedniego podpunktu
   b) poprawnie zbudowanych wyrażeń nawiasowych,
   c) dowolnej poza słowem (()).
53. Podaj gramatykę generującą język wszystkich napisów nad alfabetem {a,b}, które nie zawierają trzech liter a występujących obok siebie.
54. Podaj gramatykę generującą język wszystkich napisów nad alfabetem {a,b}, które mają parzystą liczbę liter a, ale żadne dwa a nie występują obok siebie.
55. Dana jest gramatyka G nad alfabetem {a,b} o następujących produkcjach:
   S ::= ε |aSa | bSb. Podaj gramatykę języka {a,b}* \ L(G). Wykaż jej poprawność.
56. Napisz gramatykę generującą wszystkie liczby naturalne w zapisie dwójkowym, które maja tę samą liczbę jedynek na parzystych i nieparzystych miejscach.
57. Dana jest gramatyka G nad alfabetem {0,1} o następujących produkcjach:
   A::=AA0 | 1. Napisz fragment programu sprawdzający, czy w danej tablicy A wypełnionej zerami i jedynkami zapisane jest słowo, które należy do języka tej gramatyki.
58. Podaj gramatykę dla języka nad alfabetem \a, b\, złożonego ze słów, w których liczba wystąpień symbolu a jest parzysta i żadne dwa symbole a nie występują obok siebie.
59. Wyrażenie jest zdefiniowane przez gramatykę o następujących produkcjach:
   a) <wyrażenie>::= <a>|<a><mniej><wyrażenie>
   b) <a>::=<b>|<a><więcej><b>
   c) <b>::=<c>|<c>+<b>
   d) <c>::=<liczba>|(<wyrażenie>)
   e) <liczba>::=<cyfra>|<liczba><cyfra>
   f) <cyfra>::=0|1|2|3|4|5|6|7|8|9|
   g) <mniej>::= div |-
   h) <więcej>::=*|^
   Wyznacz wartość następujących wyrażeń i zaznacz nawiasami kolejność wykonywania działań, tak aby była ona zgodna z wyprowadzeniem tych wyrażeń w podanej gramatyce:
   a) 51+23 div 6^2
   b) 2^1+3*2
   c) 20*3-(7-3-1)-7 div 2
60. Dany jest następujący fragment programu; wszystkie zmienne typu integer, x>=0)

```
a:=0; p:=1; s:=3; r:=3;
while p<=x do begin {s=3(a+1)(a+1), r=3(2a+1) ...        (*)}
  a:=a+1;
  r:=r+6;
  s:=s+r;
```

```
  p:=p+s-(r div 2)-1;
end;
```

Pokaż, że warunki na r i s podane w wierszu oznaczonym gwiazdką są niezmiennicze względem pętli programu. Które z poniższych warunków można dodać nie zaburzając niezmienniczości względem podanej pętli?

a) $p>a^2$
b) $p>a^3$
c) $p>a^4$
d) p >= ln(a+1)
e) $p \le (a+1)^2$
f) $p \le (a+1)^3$
g) $p \le (a+1)^4$

Powyższy fragment programu nadaje zmiennej a wartość pewnej funkcji zmiennej~ x. Wyznacz tę funkcję i udowodnij, że podana pętla wartość tej funkcji wylicza.

61. Dane są dwie funkcje $f,g:\{a,b\}^*\to Z$ określone w następujący sposób:

$f(\varepsilon)=0$, $f(aw)=g(w)+1$; $f(bw)=g(w)-1$

$g(\varepsilon)=0$, $g(aw)=f(w)-1$; $g(bw)=f(w)+1$ dla każdego słowa $w\in\{a,b\}^*$.

Niech $L=\{w\in\{a,b\}^*:f(w)=0\}$. Podaj gramatykę języka L.