Egzamin B **Wstęp do programowania imperatywnego** 4.03.2013

Zadanie 1

Dany jest labirynt w tablicy A:tab, gdzie tab=array[1..n,1..m] of Boolean dla n,m>0, przy czym na pole [i,j] można wejść wtedy i tylko wtedy, gdy A[i,j]=true. Z pola [i,j], na które można wejść, prowadzą krawędzie do pól [i-1,j], [i+1,j], [i,j-1], [i,j+1] – oczywiście jeśli ich indeksy mieszczą się w podanym zakresie i też można na nie wejść.
Napisz funkcję

JestDrzewo(const A:tab):Boolean;

która sprawdzi, czy tak zadany graf jest drzewem binarnym.

Zadanie 2

```
type listadrzew=^elemlistydrzew;
    elemlistydrzew = record
                      w:drzewo; nast:listadrzew
                     end;
```

Drzewo jest kompletne, jeżeli na każdym jego poziomie występują wszystkie możliwe węzły. Napisz funkcję

samekompletne (d : drzewo) : listadrzew;

która przekaże w wyniku listę wszystkich węzłów drzewa d, które są korzeniami kompletnych drzew binarnych. Węzły powinny występować na liście w kolejności prefiksowej w obiegu od lewej do prawej (tzn. jeżeli w1 poprzedza w2 w liście, to występuje wcześniej w porządku prefiksowym LP dla drzewa d niż w1).

Zadanie 3

Napisz funkcję

ilerozkładów(n:Integer):Integer;

która wyznaczy liczbę różnych rozkładów liczby n na sumę nieujemnych potęg dwójki. Na przykład ilerozkładów(8) powinno dać wartość 10, gdyż

$8 = 1+1+1+1+1+1+1+1 = 1+1+1+1+1+1+2 = 1+1+1+1+2+2 = 1+1+1+1+2^2 = 1+1+2+2+2 = 1+1+2+ 2^2 = 2+2+2+2 = 2+2+2^2 = 2^2+2^2 = 2^3$.

Zadania oddajemy na osobnych kartkach podpisane, z podanymi inicjałami prowadzącego ćwiczenia. Wszystkie rozwiązania należy uzasadnić i podać koszty wykonania algorytmów.