

Egzamin poprawkowy z programowania
część I

Zadanie 1. (15p) Zamień poniższe programy na ich odpowiedniki, w których nie ma instrukcji `goto`, `break`, `continue` oraz pętli `while`, a jedynymi strukturami sterującymi są pętla `do while` i instrukcja `if`.

- a)

```
while (b)
{
    C1;
    if (b1) break;
    C2;
    if (b2) continue;
    C3;
}
```
- b)

```
if (b1) goto L;
C1;
do {
    C2;
L:    C3;
} while (b);
```
- c)

```
do {
    C1;
L:    C2;
} while (b);
C3;
if (b1) goto L;
```

Za każdy podpunkt możesz otrzymać **(5p)**.

Zadanie 2. (16p) Będziemy używać następujących typów danych:

```
struct Item { int key; Item *next }
struct Tree { int key; Tree *left; Tree *right}
```

- a) Napisz funkcję `Item* reverse(Item *L)` zmieniającą na przeciwną kolejność elementów na liście `L` **(5p)**.
- b) Napisz funkcję `Tree* reverse(Tree *t)` zmieniającą na przeciwny porządek gałęzi w drzewie binarnym `t` **(3p)**.
- c) Drzewa o większej niż dwa liczbie potomków można reprezentować za pomocą drzew binarnych zgodnie z regułą: „na lewo syn, na prawo brat”, inaczej mówiąc do lewego wskaźnika węzła rodzica przypięta jest lista jego dzieci zbudowana z wykorzystaniem prawych wskaźników. Narysuj **(2p)** w tej reprezentacji drzewo binarne odpowiadające pełnemu drzewu trójkowemu o wysokości 2 (drzewo to ma 3 węzły na głębokości 1 oraz 9 węzłów na głębokości 2). Napisz funkcję, która odwraca drzewo (niebinarne) zapisane zgodnie z tą konwencją **(6p)**.

Zadanie 3. (12p) Niech r , s oraz t będą dowolnymi wyrażeniami regularnymi. Powiedz, które z poniższych zdań są prawdziwe¹, a które nie (**(2p)** za poprawną odpowiedź, **(-2p)** za błędną).

- a) $(r^*)^* = r^*$
- b) $(rs)^*t = r(st)^*$
- c) $(r+s)^* = (r^*s^*)^*$
- d) $r(s+t) = rs+rt$
- e) $s(r+rs)^* = (s+sr)^*r$
- f) $(r+s+rss^*(rr+ss)^*)^* = (r+s+ssr^*(sr+rs)^*)^*$

Zadanie 4. (10p) Poniżej znajduje się niedokończona procedura sortująca N -elementową tablicę T zawierającą liczby od 0 do $M-1$.

¹Przypominam, że równość wyrażeń regularnych oznacza równość definiowanych przez nie języków

```

void counting_sort(int *T, int N, int M)
{
    int B[M];           // Ilości wystąpień poszczególnych M

    for (i=0;i<M;i++)
        B[i]=0;

    for (i=0; i<N; i++)
        B[T[i]] = B[T[i]]+1;

    wsk = <a1> ;
    for ( i= <b3> )
        for (j= <c3> )
        {
            T[wsk] = <d3>;
            wsk = wsk+1;
        }
}

```

Napis <xN> oznacza, że x jest nazwą brakującego fragmentu programu, którego poprawne wstawienie da N punktów. Zatem do dzieła!

Zadanie 5. (8p) Gramatyki G_1, G_2 nad alfabetem $\{a, b\}$, określone są, odpowiednio, przez zbiory produkcji P_1, P_2 , gdzie

$$P_1 = \{S \rightarrow aS, S \rightarrow aSb, S \rightarrow bSa, S \rightarrow \varepsilon\}$$

$$P_2 = \{S \rightarrow aS, S \rightarrow SS, S \rightarrow aSb, S \rightarrow bSa, S \rightarrow \varepsilon\}$$

Pokaż, że

- (3p)** Jeżeli $w \in L(G_2)$ to $\text{waga}(w) \geq 0$, gdzie $\text{waga}(w) = |w|_a - |w|_b$.
- (3p)** $L(G_1) \neq L(G_2)$
- (2p)** G_2 jest niejednoznaczna.