

## Egzamin z programowania część I

We wszystkich zadaniach testowych obowiązuje następująca punktacja: 0 punktów za jedną lub mniej poprawnych odpowiedzi, 1 punkt za dwie poprawne odpowiedzi, 2 punkty za trzy, 5 za cztery i 10 za wszystkie poprawne odpowiedzi.

Ocena za tę część egzaminu liczona jest zgodnie z zasadą: **dst** za 35 punktów i każde dodatkowe 10 punktów powoduje podniesienie oceny o pół stopnia. Ocenę **bdb** można zatem otrzymać za 75p.

**Zadanie 1.(10p)** Rozważamy programy w języku  $D^*$  (bez procedur).  $P$  jest programem napisanym w tym języku. Czy rozstrzygalne są następujące problemy?

- a) Czy w  $P$  istnieje instrukcja, która wykona się więcej niż 1000 razy.
- b) Czy program zakończy swoje działanie wykonując mniej niż  $1000^{1000^{1000}}$  instrukcji?
- c) Czy istnieje zmienna, której zawartość zostanie przez program zmieniona?
- d) Czy zmienna  $v$  zmieni swoją wartość podczas wykonywania programu?
- e) Załóżmy, że wśród zmiennych  $P$  wybieramy pewną klasę zmiennych „sterujących”, o nazwach  $x_1, x_2, x_3, \dots$ . Mogą one być zmieniane jedynie w konstrukcji

$x_i = w; \text{ while } (x_i > 0) \{ x_i = x_i - 1; \dots \}^1.$

Ponadto wszystkie pętle **while** są takiej postaci. Problem: jaka jest wartość zmiennej  $v$  po zakończeniu działania programu?

**Zadanie 2.(10p)** W zadaniu tym rozważamy języki nad alfabetem  $\{a, b\}$ . Zmienne  $r$  oraz  $s$  oznaczają dowolne wyrażenia regularne nad tym alfabetem. Czy są prawdziwe następujące zdania?

- a) Istnieje automat skończony akceptujący język zadany przez gramatykę ze zbiorem produkcji

$$P = \{S \rightarrow aSa, S \rightarrow b, S \rightarrow a, S \rightarrow \varepsilon\}$$

- b)  $L((r^*s^*)^*) = L((r+s)^*(r+s)^*)$
- c)  $L((a+ab)^*) = L((a+b)^*) \setminus L((a+b)^*bb(a+b)^*)$
- d) Język  $\{w \mid w^R \in L(r)\}$  jest regularny.
- e) Język  $\{w_1^n w_2^n \mid w_1 \in L(r), w_2 \in L(s), n \in N\}$  jest regularny.

**Zadanie 3.(10+xp)** Czy poniższe gramatyki są jednoznaczne? Dla gramatyk niejednoznacznych wskaż słowo o dwóch drzewach wyprowadzenia i pokaż te drzewa (po dwa punkty za każdą niejednoznaczną gramatykę).

- a)  $S \rightarrow aSb \mid \varepsilon$
- b)  $S \rightarrow aSbS \mid bSaS \mid \varepsilon$
- c)  $S \rightarrow S(S)S \mid \varepsilon$
- d)  $S \rightarrow +SS \mid *SS \mid a$
- e)  $S \rightarrow S + S \mid SS \mid S * \mid (S) \mid a$

**Zadanie 4.(15p)** Zamień poniższe programy na ich odpowiedniki, w których nie ma instrukcji **goto**, a jedynymi strukturami sterującymi są pętla **do while** i instrukcja **if**.

- a) 

```
L1: C1;
    do {
        C2;
        if (b2) goto L1;
        if (b3) break;
        C3;
    } while (b1)
```
- b) 

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

---

<sup>1</sup> We fragmencie z trzema kropkami zmienna  $x_i$  służy jedynie odczytywaniu.

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

**Zadanie 5.(17p)** W punktach a) i b) możesz założyć, że typ `int` jest wystarczająco pojemny i nie trzeba kontrolować przekroczenia zakresu.

- Napisz funkcję `void drukuj(int n)` wypisującą w postaci binarnej dodatnią liczbę `n` (5p).
- Napisz funkcję `int bin(char*)` przekształcającą napis złożony z zer i jedynek na odpowiednią liczbę całkowitą (5p).
- Rozważmy gramatykę z produkcjami:

$$S \rightarrow 11 \mid 1001 \mid SS \mid S0$$

Pokaż, że wszystkie słowa z  $L(G)$  potraktowane jako liczby binarne są podzielne przez 3. Czy każda liczba binarna o tej własności należy do  $L(G)$ ? (odpowiedź uzasadnij) (6+2p).

**Zadanie 6.(26p)** W tym zadaniu wyrażenia arytmetyczne zawierają jednoliterowe zmienne, nawiasy oraz znaki `*`, `+` (z tradycyjnym priorytetem działań i łącznością). Nie zawierają żadnych spacji. Będziemy je reprezentować za pomocą następujących struktur:

```
struct It { Ex *v; It* next;}
struct Ex { char c; It* body;}
```

W wyrażeniu pamiętamy znak (nazwę zmiennej lub znak arytmetyczny). Dla znaku arytmetycznego pamiętamy listę jego argumentów (ponieważ operacje `+` oraz `-` są łączne, argumentów może być więcej niż 2).

- Uzupełnij program, który czyta ciąg znaków reprezentujący poprawne wyrażenie arytmetyczne i zwraca wskaźnik na strukturę `Ex` reprezentującą to wyrażenie.

```
// Wyrażenie znajduje się w globalnej zmiennej char *W;
// Funkcje używają też globalnego indeksu do tej tablicy - zmiennej Poz
// (początkowo równej 0)
Ex* Wyrazenie()
{
    int k=0;
    Ex* e = Skladnik();
    It* wsk;
    while (W[Poz] == '+') {
        Poz++;
        Ex* e2 = Skladnik();
        if (k == 0) {
            Ex *E = new Ex; wsk = new It;
            E->c = (1);
            E->body = (2);
            wsk->v = (3);
            k=1;
        }
        wsk->next = (4);
        wsk = (5);
        wsk->v = (6);
    }
    wsk->next = 0;
    if (k==0) return (7);
    return (8);
}
```

```
Ex Skladnik() {
    // tresc analogiczna do wyrazenia, wywołuje procedure czynnik
```

```
}
```

```
Ex* Czynn timer()
```

```
{
```

```
    if (is_letter(W[Poz])) {
```

```
        (9) // za 2 p.
```

```
    }
```

```
    else if (W[Poz] == '(' ) {
```

```
        (10) // za 2 p.
```

```
    }
```

```
    else throw Bled();
```

```
}
```

Przyporządkuj fragmenty kodu fragmentom oznaczonym przez (1), (2), (3) itd. Jeżeli nie jest napisane inaczej to każda prawidłowa odpowiedź jest warta 1 punkt.

b) Napisz funkcję `Drukuj(Exp *E)`, która drukuje w postaci infiksowej wyrażenie E. Możesz założyć że każda lista w strukturze `Exp` ma co najmniej dwa argumenty **5p**.

c) Napisz funkcję `BezNawiasów(char *)`, która usuwa niepotrzebne nawiasy w napisie stanowiącym wyrażenie i korzysta z jego reprezentacji w postaci struktury `Exp`. Przykładowo

```
BezNawiasów("((a+b)+((c*d))+e*(a+(b)))")
```

powinien wypisać

```
a+b+c*d+e*(a+b).
```

Za tę część można otrzymać **10p**.

**Zadanie 7.(12p)** Rozważmy poniższy program:

```
void f(int x) {
```

```
    int i=4;
```

```
    cout << x << " ";
```

```
    a[0]=8; i++;
```

```
    cout << x << " ";
```

```
    x=x+1;
```

```
}
```

```
void main() {
```

```
    int a[]= {0,1,2,3,4,5,6,7};
```

```
    int i=0;
```

```
    f(a[i++]);
```

```
    cout << a[0] << endl;
```

```
}
```

Program ten wypisuje 3 liczby. Podaj te liczby przy założeniu, że:

- a) parametr `x` jest przekazywany przez wartość,
- b) parametr `x` jest przekazywany przez wartość i wynik,
- c) parametr `x` jest przekazywany przez nazwę,
- d) parametr `x` jest przekazywany przez referencję.