

PROGRAMIRANJE ZA ELEKTROTEHNIKE II

Rešitev 1. testa, 15. 4. 2009

1. (2) Napišite, kaj je sintaksa programskega jezika.

Sintaksa programskega jezika je niz pravil, ki predpisujejo pravilno zaporedje simbolov v zapisu v tem jeziku.

2. (2) Pojasnite, katerim spremenljivkam pravimo lokalne, katerim pa globalne.

Lokalne spremenljivke so tiste, ki so definirane znotraj neke funkcije, globalne pa tiste, ki so definirane izven vseh funkcij.

3. (9) Definirajte strukturo z imenom `vrata`. Radi bi jo uporabili za predstavitev vrste logičnih vrat. Zato naj ima dva elementa. Prvi element naj pove, ali gre za vrata AND, OR ali XOR, drugi element pa naj pove število vhodov vrat. Prvi element naj bo naštevnega tipa z elementi `vrataAND`, `vrataOR` in `vrataXOR`, ki ga poprej definirajte. Drugi element naj bo nepredznačeno kratko celo število.

```
enum vrsta {vrataAND, vrataOR, vrataXOR};
```

```
struct vrata {  
    vrsta vrstaVrat;  
    unsigned short stVhodov;  
};
```

4. (2) Napišite definicijo spremenljivke `vrata1` tipa `vrata` iz prejšnjega vprašanja.

```
vrata vrata1;
```

5. (1) Pojasnite, v čem je osnovna razlika med spremenljivkami tipa `float`, `double` in `long double`.

Osnovna razlika je v številu zlogov, ki se uporabljajo za hranjenje vrednosti spremenljivke.

6. (2) V istem programu lahko imamo več funkcij z enakim imenom. V čem pa se morajo razlikovati, da je to dovoljeno?

Razlikovati se morajo v tipu ali številu argumentov.

7. (14) Natančno napišite, kaj izpiše naslednji program. Nato napišite še vsebino spremenljivke `polje` tik pred izvedbo stavka `return`.

```
#include <iostream>
using namespace std;
int main(){
    const int velikost = 4;
    int polje[] = {1, 2, 3, 4};
    int vsota = 0;
    int stevec = 0;

    do {
        vsota += polje[stevec++];
        if (stevec % 2 == 0) {
            cout << "Vsota: " << vsota << endl;
            cout << "Stevec: " << stevec << endl << endl;
        }
        else {
            polje[stevec - 1] = 0;
        }
    } while (stevec < velikost);

    return 0;
}
```

Program izpiše:

Vsota: 3
Stevec: 2

Vsota: 10
Stevec: 4

Vsebina spremenljivke `polje` na koncu je {0, 2, 0, 4}.

8. (12) Napišite definicijo funkcije z imenom `poisciMin`. Naj bo celoštevilčnega tipa in ima dva formalna argumenta. Prvi naj bo polje poljubne dolžine, katerega elementi so cela števila, drugi pa dolžina tega polja. Funkcija naj poišče in vrne najmanjši element tega polja. Funkcijo bi radi uporabili v glavnem programu, kjer bi najprej zaslona prebrali dolžino polja, v katerem hočemo poiskati najmanjši element, potem pa to polje. Potem bi s tem poljem in dolžino kot dejanskima argumentoma poklicali funkcijo `poisciMin`, da bi nam vrnila najmanjši element. Glavnega programa ni treba napisati!

Na primer:

```
int poisciMin(int polje[], int dolzina) {
    int minVred = polje[0];
    for (int i = 1; i < dolzina; i++) {
        if (polje[i] < minVred)
            minVred = polje[i];
    }
    return minVred;
}
```

9. (20) Napišite delujoč program, ki bo za uporabnika izračunal in izpisal na zaslon fakulteto vtipkanega naravnega števila (lahko se vtipka tudi 0), dokler bo uporabnik to hotel. Vemo, da je fakulteta $n!$ za $n = 0$ enaka 1, sicer pa je $n! = n * (n-1) * (n-2) * \dots * 1$.

Glavni program (tj. funkcija `main()`) naj računa fakulteto s klicem funkcije, ki jo definirajte pred glavnim programom. Naj bo celoštevilčnega tipa, edini formalni argument pa naj bo celo število n , za katero naj vrne izračunano fakulteto. (Funkcijo za izračun fakultete lahko napišete z uporabo rekurzije, če menite, da to znate.)

Glavni program naj deluje tako. Ko uporabnik požene program, naj ga glavni program prosi za vnos vrednosti n prek tipkovnice. Program prebere vneseno vrednost ter izračuna in izpiše na zaslon njeno fakulteto. Potem uporabnika vpraša, ali želi nadaljevati ali končati. Če uporabnik vtipka znak 'D', pomeni, da želi nadaljevati in program ga spet vpraša za vrednost n ter izračuna in izpiše fakulteto. Če uporabnik vtipka znak 'N', pa se program konča. Program naj jemlje vrednost n in izračuna ter izpiše fakulteto za uporabnika ter ga vpraša, ali želi nadaljevati, tako dolgo, dokler uporabnik ob vprašanju o nadaljevanju ne vtipka 'N'.

Na primer:

```
#include <iostream>
using namespace std;

int fakulteta(int n) {
    int rezultat = 1;
    for (int i = n; i > 0; i--)
        rezultat *= i;
    return rezultat;
}

int main() {
    char nadalj;
    int n;
    do {
        cout << "Vpisi vrednost n:" << endl;
        cin >> n;
        cout << "Fakulteta je: " << fakulteta(n) << endl;
        cout << "Ali zelis nadaljevati [D/N]?" << endl;
        cin >> nadalj;
    } while (nadalj == 'D');
    return 0;
}
```