

## Ispit iz Programiranja 2

Ispit traje 120 minuta

### Napomene:

- a) Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.  
 b) Vrednost odgovora: tačan = **5**; netačan = **-1.25**; nevažeci (nula ili više zacrtnjenih kružića) = **0**.  
 c) Na pitanjima se može osvojiti najviše **15** poena. Prvi zadatak nosi **25** poena, dok drugi nosi **30** poena.

### I ZADACI

**1)** Napisati program na programskom jeziku C koji pronalazi najfrekventnije slovo u dinamičkoj matrici slova engleske abecede. Svaka vrsta ove dinamičke matrice sadrži jednu reč, učitane iz posebnog reda, a maksimalna dužina jedne reči je 20 slova. Broj reči nije poznat unapred, a unos reči prestaje kada se učita prazan red. Potrebno je pročitati sa standardnog ulaza reči i smestiti ih u dinamičku matricu, tako da reči ne zauzimaju više prostora nego što je potrebno. Nakon toga, potrebno je pronaći i ispisati slovo koje se najviše puta pojavljuje u formiranoj matrici. Veliko i njemu odgovarajuće malo slovo treba tretirati na isti način. Ukoliko se više različitih slova pojavljuje isti broj puta, ispisati ih sve. Učitavanje reči i obradu realizovati kao zasebne potprograme koji sa glavnim programom komuniciraju isključivo putem argumenata i povratnih vrednosti. Voditi računa o korektnoj upotrebi dinamičke memorije i optimalnom iskorišćenju memorijskog prostora.

**2)** Napisati program na programskom jeziku C koji pronalazi najjeftinije komponente u cenovnom rangu. Komponente su zadate u tekstualnoj datoteci `komp.txt`. Svaka komponenta je zadata u jednom redu datoteke po sledećem formatu: tip (može biti CPU, MEM ili BOARD), ime komponente (jedna reč sa maksimalno 30 znakova) i cena komponente (realan broj). Program treba da pronađe tačno jednu komponentu za svaki tip koja je najjeftinija od 20% najskupljih komponenti tog tipa kao kompromis između cene i kvaliteta i da ispiše ime te komponente na standardnom izlazu. Program treba i da ispiše ukupnu cenu svih odabranih komponentata. Zaokruživanje se vrši na prvi veći ceo broj (npr. ako ima 6 komponenti, u 20% najskupljih se nalaze 2 komponente). Broj komponentata u datoteci nije poznat i treba voditi računa o ispravnoj upotrebi memorijskih resursa.

### II PITANJA

- 1)** Koja od sledećih tvrdjenja su tačna za programski jezik C?  
 A) Ukoliko je definisano `int a[]={1,2,3}`, `*const pa=a`; dozvoljeno je izvršiti dodelu vrednosti `pa=a+5`;  
 B) Ukoliko je `int *p` pokazivač u koji je upisana adresa koju je vratila funkcija `malloc`, poziv funkcije `free(p+5)` će biti korektno izvršen;  
 C) Ukoliko je definisano `int x=1`, `*px=&x`; `void *pv=px`; dozvoljeno je izvršiti dodelu vrednosti `*(int*)pv=*px+5`;

**2)** Šta ispisuje sledeći program napisan na programskom jeziku C ako se program pokrene pomoću komande `program.exe 90 100 5 6`?

```
#include <stdio.h>
#define PART 10
typedef unsigned long data_t;
data_t mul(data_t a, data_t b) {return a * b;};
data_t div(data_t a, data_t b) {return a / b;};
data_t add(data_t a, data_t b) {return a + b;};
data_t rd(data_t a, data_t b) {
    if (a & (1 << (PART - b))) {
        return a + (1 << (PART - b));
    }
    return a;
}

int main(int argc, char *argv[]) {
    int i; data_t a = 25;
    a <<= PART;
    a = mul(a, atoi(argv[1]));
    a = div(a, atoi(argv[2]));
    a = rd(a, 1);
    a = add(a, atoi(argv[4]));
    printf("%lu.%lu",
           a >> PART, a & ~(~0 << PART));
    return 0;
}
```

- (A) 23.6                      B) 27.5                      C) 29.0

**3)** Šta radi sledeća funkcija napisana na programskom jeziku C?

```
int f(char* string, char* chars) {
    char c, *p, *s;
    for (s = string, c = *s; c != 0; s++, c = *s)
        for (p = chars; *p != 0; p++)
            if (c == *p) return s-string;
    return s-string;
}
```

A) Pronalazi prvu poziciju poslednjeg karaktera iz stringa `chars` koji se pojavi u stringu `string`. Pozicije se označavaju počev od nule.

B) Pronalazi dužinu početnog segmenta u stringu `string` koji se sastoji samo od znakova koji nisu u stringu `chars`.

C) Pronalazi dužinu krajnjeg segmenta u stringu `string` koji se sastoji samo od znakova koji su u stringu `chars`.

**4)** Šta ispisuje sledeći program na programskom jeziku C ukoliko su u jednostruko ulančanu listu `lst` redom uneseni brojevi 2 1 8 5 4 3? Smatrati da funkcija `citaj` ispravno formira, a funkcija `pisi` ispravno ispisuje sadržaj zadate liste redom od početka.

```
#include <stdio.h>
#include <stdlib.h>
typedef struct elem {
    int broj; struct elem *sled;
} Elem;
typedef Elem* (*FP)(Elem *, int);
Elem *f1(Elem *lst, int b);
Elem *f2(Elem *lst, int b);
void main () {
    Elem *lst, *lst1, *lst2 = NULL;
    FP fun[] = { f2, f1 };
    lst = citaj(); lst1 = lst;
    while (lst1 != NULL) {
        lst2 = fun[lst1->broj & 1]
                (lst2, lst1->broj);
        lst1 = lst1->sled;
    }
    pisi(lst2); brisi(lst); brisi(lst2);
}
```

```
Elem *f1(Elem *lst, int b) {
    if (!lst) {
        lst = malloc(sizeof(Elem));
        lst->broj = b; lst->sled = NULL;
    } else lst->sled = f1(lst->sled, b);
    return lst;
}
Elem *f2(Elem *lst, int b) {
    if (lst) {
        if (lst->broj != b) {
            lst->sled = f2(lst->sled, b);
        }
        else {
            Elem *stari = lst;
            lst = f2(lst->sled, b); free(stari);
        }
    }
    return lst;
}
```

- (A) 153                      B) 135                      C) 284