

Ispit iz Programiranja 2

Trajanje: **150** minuta

Napomene:

- a) Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.
- b) Vrednost odgovora: tačan = **5**; netačan = **-1.25**; nevažeći (nula ili više zacrnjenih kružića) = **0**.
- c) Na pitanjima se može osvojiti najviše **25** poena.
- d) Zadaci nose po **20** poena.

I. ZADACI

1) Dat je neuređeni niz celih brojeva u kome se elementi mogu ponavljati više puta. Napisati funkciju na jeziku C koja transformiše niz na sledeći način: u nizu se pojavljuju parovi (element, broj pojava), bez ponavljanja elemenata, pri čemu redosled elemenata odgovara redosledu prvog pojavljivanja u nizu. Na primer, niz 1 3 0 1 5 3 1 5 5 1 7 biće transformisan u 1 4 3 2 0 1 5 3 7 1, jer se jedinica pojavljuje 4 puta, trojka 2 puta, nula jednom, petica tri puta i sedmica jednom). Pri transformaciji se formira novi niz, a prvobitni ostaje sačuvan. Maksimalna dužina početnog niza je 100 elemenata. Napisati glavni program koji poziva ovu funkciju sve dok se za dužinu niza ne unese nula.

2) Napisati program na programskom jeziku C koji određuje broj pruženih usluga i računa bruto zaradu na godišnjem nivou koji je ostvarila privatna stomatološka klinika. Podaci o pruženim uslugama smešteni su u tekst datoteku **medicus.txt** prema sledećem formatu: u svakom redu datoteke nalazi se zapis o jednoj usluzi, tako što se najpre navodi redni broj meseca i dana kada je usluga pružena a zatim šifra usluge. Svi podaci su celi brojevi. Cena usluge smeštena je u tekst datoteku **sifre.txt** koja u svakom redu sadrži sledeće podatke: šifra usluge (ceo broj) i cena usluge (realan broj). Broj pruženih usluga nije poznat unapred, međutim poznato je da je broj vrsta usluga (različitih šifara) između 1 i 100. Na osnovu podataka iz opisanih datoteka, program treba da odredi i na standardnom izlazu ispiše bruto zaradu cele klinike na godišnjem nivou, a zatim, za svaku uslugu, koliko puta je pružena u toku cele godine.

II. PITANJA

1) Ako postoji deklaracija: `char *s, *r;` koja od sledećih naredbi `for` obezbeđuje pravilno nadovezivanje stringa na koji pokazuje `r` na string na koji pokazuje `s`? Smatrati da je unapred alocirano dovoljno memorije za smeštanje stringova.

- (A)** `for(;*r;*s++ ? 0 : (*(s-1)=*r++, *s=0));`
- B)** `for(*s;*s++ ? 0 : *--s = *r++, *++s=0);`
- C)** `for(*r;*s++ ? 0 : *s = *r++, *++s=0);`

2) Dat je deo programskog koda na jeziku C za izbacivanje nekih elemenata iz jednostruko ulančane liste celih brojeva. Funkcija oslobodi treba da iz liste izbaci sve one elemente liste čije polje `num` ima vrednost `broj`, koja se zadaje kao argument funkcije. Šta treba da stoji na mestu ##### da bi ova funkcija ispravno obavljala tu operaciju?

```
#include <stdlib.h>
#include <stdio.h>
typedef struct elem { int num; struct elem *sled; } Elem;

Elem *oslobodi(Elem *e, int broj){
Elem *tmp;
    if( e == NULL ) return NULL;
    tmp = oslobodi(e->sled, broj);
    #####
    e->sled = tmp; return e;
}
```

- A)** `if(e==tmp) { free(e); return tmp; }`
- B)** `if(e->num==broj) { free(e); return e; }`
- (C)** `if(e->num==broj) { free(e); return tmp; }`

3) Šta ispisuje sledeći program na programskom jeziku C?

<pre>#include <stdio.h> int f0(int x) {return (x*7+2)%6;} int f1(int x) {return (x+13)%6;} int f2(int x) {return (x*5)%6;} </pre>	<pre>void main(){ int s = 0, i = 0, j; int (*f[6])(int) = {f0, f2, f1, f2, f1, f0}; for (j=0; j < 3; j++) s += i = (*f[i])(i); printf("%d",s); }</pre>
---	---

A) 5

B) 10

(C) 8

4) Koje od ponuđenih tvrdnji su tačne za programski jezik C?

(A) Dozvoljeno je sabiranje pokazivača i proizvoljnog celog broja;

B) Povratni tip funkcije `main()` uvek mora biti `void`;

(C) Ispravna je sledeća deklaracija: `char *(*f)(char **);`

5) Šta će ispisati sledeći program na programskom jeziku C ako se program nalazi u fajlu `program.exe` i ako mu se prilikom pokretanja proslede sledeći parametri: `programiranje mikro mikroprocesor mikroproces mikron`?

<pre>#include <stdio.h> void main(int argc, char* argv[]){ char *p; int i, j = 0, k = 0; for (i = 2; i < argc; i++){ p = argv[i-1]; </pre>	<pre> for(j = 0; argv[i][j] && *p; j++, p++) if (*p != argv[i][j]) break; if (!*p !argv[i][j]) k = k>j?k:j; } printf("%d",k); }</pre>
---	--

(A) 11

B) 4

C) 13

6) Za $n \leq 10$, potprogram `xyz` radi sledeću operaciju nad kvadratnom matricom a dimenzija $n \times n$:

```
void xyz(int **a, int n) {
    int t,i,j;
    for (i=0; i<n; i++)
        for (j=0; j<n; j++)
            t=*(a[i]+j),
            *(a[i]+j)=*(*(a+j)+i),
            a[j][i]=t;
}
```

A) Zamenjuje mesta elementima simetričnim u odnosu na glavnu dijagonalu;

(B) Ne menja sadržaj matrice;

C) Zamenjuje mesta elementima simetričnim u odnosu na sporednu dijagonalu;