

Ispit iz Programiranja 2

Ispit traje 150 minuta.

Napomene:

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

I ZADACI

1) Napisati program na programskom jeziku C koji vrši obradu nad kvadratnom matricom dimenzije NxN ($N \leq 20$). Program najpre treba da učitava N, a zatim i elemente matrice, tako da je svaka vrsta u posebnom redu. Nakon učitavanja, program treba proveriti da li je element na poziciji (i, j) iznad glavne dijagonale veći od elementa na poziciji (j, i) ispod glavne dijagonale matrice i ukoliko jeste, zameni im mesta. Nakon obrade, ispisati rezultujuću matricu na standardni izlaz. Obradu matrice realizovati u potprogramu koji sa glavnim programom komunicira isključivo pomoću argumenata i povratne vrednosti.

2) U datoteci **pacijenti.txt** nalazi se spisak pacijenata jedne zubarske ordinacije. Svaki red datoteke sadrži sledeće podatke o pacijentima: celoborjni identifikator pacijenta, a zatim ime i prezime pacijenta. Datoteka **intervencije.txt** sadrži spisak intervencija. Svaki red sadrži identifikator pacijenta, datum intervencije u formatu **dd.mm.yy.**, zatim oznaku vilice na kojoj se nalazi zub (D za donju, G za gornju), oznaku strane na kojoj je zub (L za levo, D za desno), a potom i redni broj zuba (cifra od 1 do 8), i nakon toga cenu intervencije. Moguće je da jedna osoba ima više od jedne intervencije nad jednim zubom. Napisati program na programskom jeziku C koji prvo učitava potrebne podatke iz navedenih datoteka, potom učitava oznaku jednog zuba (npr. „DL5“ - dole levo 5), a onda pronađe korisnika koji je imao najviše intervencija na zadatom zubu, i ispiše njegov identifikator, ime, prezime i ukupnu sumu novca koji je dao na popravku tog zuba na standardni izlaz.

II PITANJA

1) Koji od sledeća tri segmenta koda ne sadrži greške i pritom ispisuje vrednost 1? Smatrati da su pri prevođenju uključena sva potrebna standardna zaglavlja.

I	II	III
<pre>void f(){ int a[10]; int* b = a; void* k = b; b++; k = b; k++; printf("%d", k-b); }</pre>	<pre>void g(){ char a[10]; char *b = malloc(10); printf("%d", sizeof(a) == sizeof(b)); }</pre>	<pre>void h(){ char s1[] = "Ospiti", s2[] = ""; s1[0]='I'; s2 = "Ispiti"; printf("%d", !strcmp(s1,s2)); }</pre>

A) I

B) II

C) III

2) Šta ispisuje sledeći program na programskom jeziku C?

```
#include<stdio.h>
int fun(int n){ return n ? (n&1) + fun(n>>1) : n; }
int main(){
    int a = fun(3), b = fun(7), c = fun(64);
    printf("%d", a + b + c); return 0;
}
```

(A) 6

B) 74

C) 5

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

- Dozvoljeno je definisati niz funkcija, ukoliko funkcije nemaju formalne argumente.
- Ako su **s1** i **s2** strukturne promenljive istog tipa, dozvoljena je dodela vrednosti **s1 = s2**;
- Polje strukture ne sme biti definisano kao pokazivač na tu istu strukturu.

4) Šta ispisuje sledeći program na programskom jeziku C ukoliko jednostruko ulančana lista sadrži redom brojeve 4 2 1 7 3 6? Smatrati da funkcija **citaj** ispravno formira, a funkcija **psi** ispravno ispisuje sadržaj jednostruko ulančane liste redom od početka.

```
typedef struct elem {
    int broj; struct elem *sled;
} Elem;
Elem* obradi (Elem* lst) {
    Elem* preth = NULL, *lst1 = NULL,
    *tek = lst, *p;
    int broj;
    if (tek) {
        broj = tek->broj;
        preth = tek; tek = tek->sled;
    }
    while (tek) {
        if (broj > tek->broj) { p = tek;
            tek = tek->sled; preth->sled = tek;
            p->sled = lst1; lst1 = p;
        } else { tek = tek->sled; }
    }
    return lst1;
}
void main () { Elem *lst, *lst1;
    lst = citaj(); lst1 = obradi(lst); psi(lst1);
}
```

(A) 3 1 2

B) 2 1 3

C) 6 7 4

5) Šta ispisuje sledeći program napisan na programskom jeziku C?

```
#include<stdio.h>
int func(const char *s1, const char *s2){
    char *sc1 = s1, *sc2;
    while (*sc1) { sc2 = s2;
        while (*sc2 && *sc2 != *sc1) sc2++;
        if (*sc2 == *sc1) return sc1 - s1;
        sc1++;
    }
    return -1;
}
```

A) 3

(B) 1

C) 4

6) Šta ispisuje dati program, čija je izvršna verzija u datoteci **program.exe**, ukoliko se pokreće preko komandne linije sledećom komandom: **program.exe 5 3 1 7** ?

```
#include <stdio.h>
#include <stdlib.h>
void main(int argc, char *argv[]) {
    int x, y, j, s = 0;
    x = atoi(argv[1]);
    for (j = argc - 1; j > 1; ) {
        y = atoi(argv[j--]);
        s += x > y ? x : y;
    }
    printf("%d %d %d", x, y, s);
}
```

A) 7 1 15

B) 3 1 18

(C) 5 3 17