

Ispit iz Programiranja 2

Ispit traje 150 minuta.

Napomene:

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

I ZADACI

1) Na nekom internet portalu korisnici mogu da ocenjuju restorane po 5 različitih kriterijuma. Svaki restoran opisan je nizom ocena (realni brojevi). Broj restorana N se učitava sa standardnog ulaza, pri čemu važi $3 \leq N \leq 20$. Podaci o ocenama se nalaze u matrici `restorani`, za svaki od N restorana u posebnoj vrsti. Prilikom ocenjivanja, korisnik zadaje i lične težinske faktore (realni brojevi) koji se smeštaju u nizu `tezine` i predstavljaju bitnost odgovarajućeg kriterijuma za tog korisnika. Potrebno je napisati potprogram koji na osnovu zadate matrice ocena i zadatog niza težinskih faktora nekog korisnika računa niz sumarnih ocena za svaki od restorana. Sumarna ocena za restoran računa se kao suma proizvoda svake od ocena po odgovarajućem kriterijumu i vrednosti odgovarajućeg težinskog faktora. Napisati glavni program na programskom jeziku C koji učitava dimenzije i samu matricu `restorani`, a zatim i niz `tezine` za nekog korisnika. Nakon učitavanja, program treba da pozove realizovani potprogram za računanje ocena i ispiše ocene tri restorana sa najnižom sumarnom ocenom.

2) Napisati na programskom jeziku C program koji nalazi korisnike koji nisu na vreme izmirili račune za električnu energiju. U tekstualnoj datoteci `racuni.txt` nalaze se jedinstveni zapisi o računima korisnika električne energije, gde svaki red sadrži celobrojni identifikator brojila i iznos koji treba uplatiti. U tekstualnoj datoteci `uplate.txt` nalaze se podaci o uplatama električne energije, pri čemu svaki red sadrži identifikator brojila i iznos koji je uplaćen. Ništa se ne može pretpostaviti o broju i redosledu računa i uplata u datotekama. Može se desiti da za neke račune ne postoji nijedna uplata, a takođe je moguće izvršiti proizvoljno mnogo uplata za jedan račun. Sve uplate odnose se na postojeće identifikatore brojila. Program najpre treba da učitava podatke o svim računima u jednostruko ulančanu listu, a zatim da čita jednu po jednu uplatu iz datoteke `uplate.txt` i ažurira iznos duga za pročitani račun. Na kraju program treba da formira izlaznu tekstualnu datoteka `duznici.txt` u kojoj svaki red sadrži identifikator brojila i iznos duga (negativan broj). Voditi računa o korektnoj upotrebi zauzetih resursa.

II PITANJA

- 1) Koja od sledećih tvrdnji važe za programski jezik C?
 A) Struktura `struct s1` može sadržati polje tipa `struct s1`.
 B) Naredba `return` mora da sadrži izraz čiju vrednost funkcija vraća.
 C) U programskom jeziku C je podržan prenos argumenata funkcije isključivo po vrednosti.

2) Šta treba da stoji umesto `####` i `&&&&` da bi sledeća funkcija ispravno dodavala element u opadajuće uređenu jednostruko ulančanu listu, uz pretpostavku da alokacija memorije uvek uspeva?

```
#include <stdio.h>
#include <stdlib.h>
typedef struct elem { int broj; struct elem *sled; } Elem;
Elem *umetni(Elem *lst, int b) {
    Elem *novi = (Elem*) malloc (sizeof(Elem)), *pret = NULL;
    novi->broj = b; novi->sled = NULL;
    if (!lst) return novi;
    else { Elem *tek = lst;
        while (####) pret = tek, tek = tek->sled;
        if( !pret ) novi->sled = lst, lst = novi;
        else if( !tek ) &&&&;
        else novi->sled = pret->sled, pret->sled = novi;
        return lst;
    }
}
```

- (A) `####: tek && tek->broj>b`
`&&&&: pret->sled=novi`
 (B) `####: tek->broj>b`
`&&&&: pret->sled=NULL`
 (C) `####: tek && tek->broj > b`
`&&&&: novi->sled=pret`

3) Šta ispisuje sledeći program na programskom jeziku C, čija se izvršna verzija nalazi u datoteci `januar2014.exe`? Program je pokrenut sledećom komandom: `januar2014 januar deveti`.

```
#include <stdio.h>
void main(int argc, char* argv[]) {
    int i = 1, n = argc;
    for (; i<argc; i++) {
        putchar((argv[i])[i]);
        n += strlen(argv[i]);
    }
    printf("%d\n", n);
}
```

- A) jav22 (B) av15 (C) jd12

4) Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
char *f(char* a, char* b, int c){
    char *x = a;
    while ((*a++ = *b++) && (--c));
    *a = '\0'; return x;
}
#define SIZE 30
void main() {
    char a[] = "IspitJan2013";
    char b[SIZE];
    printf ("%s", f(b, a, 10));
}
```

- A) IspitJan201 (B) IspitJan2013 (C) IspitJan20

5) Koja od navedenih funkcija ispravno nadovezuje znakovni niz **A** na kraj znakovnog niza **B** ukoliko se niz **B** završava istim znakom kojim niz **A** počinje? Smatrati da nizovi nisu prazni i da je za niz **B** rezervisano dovoljno prostora tako da nadovezivanje može bezbedno da se izvede.

```
void n1(char *A, char *B){
    while(*B) B++;
    if(*(B-1) == *A)
        while(*B++=*A++);
}
void n2(char *A, char *B){
    while(*B++);
    if(B[-1] == *A){
        while(*A) *B++ = *A++;
        *B='\0';
    }
}
#include <string.h>
void n3(char *A, char *B) {
    int i=0, x=strlen(B)-1;
    if(B[x] == A[0])
        while(A[i]) B[x++]=A[i++];
    A[i]='\0';
}
```

- (A) n1 (B) n2 (C) n3

6) Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
void main() {
    int niz[10], *a, *b, *c, i;
    a=c=niz; b=a+9;
    for(i=0; i < sizeof(niz)/sizeof(int);i++) a[i]=i;
    while(a++ < b--) if(*b&1){ int c; c=*a; *a=*b; *b=c;}
    for(i=0;i<10;i++) printf("%d",*(c+i));
}
```

- (A) 0173456289 (B) 9123546780 (C) 0123456789