

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) Napisati program na programskom jeziku C koji sa standardnog ulaza učitava niz celih pozitivnih brojeva, nad njima izvrši obradu i rezultat ispiše na standardni izlaz. Broj brojeva nije unapred poznat, ali je potrebno korisnika ograničiti da ne unese više od 100 brojeva (ako korisnik unese više od 100 brojeva, ispisati poruku o grešci i izaći iz programa). Korisnik završava unos tako što unese vrednost manju ili jednaku nuli. Obrada se sastoji u tome da se iz niza izbaci druga vrednost po veličini. Ukoliko postoji više istih vrednosti, izbaciti ih sve. U zadatku je dozvoljeno koristiti samo jedan niz celih brojeva kapaciteta 100 elemenata.

Primer:

Ulaz: 2 15 14 20 25 20 19 25 8 0

Izlaz: 2 15 14 25 19 25 8

2) Napisati program na programskom jeziku C koji sa standardnog ulaza učitava dimenziju kvadratne matrice, zatim učitava matricu, izračuna kvadrat učitane matrice i ispiše rezultujuću matricu u izlaznu datoteku `izlaz.txt`. Vrste matrice ispisivati u zasebnim redovima i ostavljati jedan prazan red između uzastopnih vrsta. Glavni program treba da ponavlja postupak sve dok korisnik za dimenziju matrice ne unese nulu. Unos, obradu i ispis matrice realizovati kao dve nezavisne funkcije (unos jedna, obrada i ispis druga). **Napomena:** komunikacija između glavnog programa i funkcija, kao i između funkcija treba da se obavlja isključivo preko argumenata i povratnih vrednosti. Voditi računa o uspešnosti alokacije resursa i njihovom pravilnom oslobađanju.

II. PITANJA

1) Koja od sledećih tvrdjenja su tačna na programskom jeziku C?

A) Naredba `return` je obavezna u funkciji.

B) Brojačku promenljivu `for` petlje nije moguće menjati u telu `for` petlje.

(C) Ako su `s1` i `s2` pokazivači na dva znakovna niza (stringa), drugi niz se ne može nadovezati na prvi naredbom `s1+=s2`.

2) Šta ispisuje sledeći program na programskom jeziku C?

<pre>#include <stdio.h> main(){ int n[]={5,4,3,2,1,0}; int *p = n+4, a = 0, s = 0; while (a < 10) { switch (*p) { case 0: p-=*(p-2); break; case 1: a++,p--; break; case 2: p--;</pre>	<pre> case 3: a++;p-=*(p+1);break; case 4: p -= *(p-2); continue; case 5: a -= 3, p += *p; } s += a += *p; } printf("%d\n",s); }</pre>
---	---

A) 32

(B) 40

C) 12

3) Šta izračunava dati potprogram na programskom jeziku C?

<pre>int fun(unsigned int x){ unsigned int a = 0, b = 0; while (x) { if (x%2 == a) { b += a; a = !a;</pre>	<pre> } x >>= 1; } return b;}</pre>
--	--

A) Računa broj pojavljivanja sekvence 01 u binarnoj predstavi parametra x.

(B) Računa broj pojavljivanja sekvence 10 u binarnoj predstavi parametra x.

C) Računa broj pojavljivanja cifre 1 u binarnoj predstavi parametra x.

4) Šta ispisuje sledeći program na programskom jeziku C ako se pri pozivu iz komandne linije proslede sledeće vrednosti: 0x54B 0x338 Beograd Subotica Zrenjanin?

#include <stdio.h> main (int argc, char *argv[]) { unsigned int s, i; sscanf(argv[1], "%x", &s);	for (i=0; s; i++) { if(s%2) printf("%c", argv[3][i]); else printf("%c", argv[4][7-i]); s >>= 2; }}
---	---

(A) Bcigra

B) inbjt

C) Beotica

5) Šta ispisuje sledeći program na programskom jeziku C?

#include<stdio.h> int a[4][4] = { {1,2,3,4}, {5,6,7,8}, {9,10,11,12},{13,14,15,16} }; int f(int x, int y){ if (y >= 3) return a[y][x];	else if (x+y >= 3) return a[y][x] + f(x-1,y); else f(x,y+1); } main () { printf("%d",f(3,0)); }
--	--

A) 35

(B) 34

C) 52

6) Koja od datih funkcija na programskom jeziku C iz neprazne liste briše sva pojavljivanja zadate vrednosti? Parametar `l` pokazuje na početak liste, a parametar `val` je vrednost koja se izbacuje iz liste. Povratna vrednost je pokazivač na početak obrađene liste. Element liste je tipa strukture `elem`, čije polje `s1` pokazuje na sledeći element u listi, a polje `v` sadrži vrednost elementa.

A) `elem *f(elem *l, int val){
 elem *p = l->s1, *e;
 e = l;
 while (p){
 if (p->v == val) {
 l->s1 = p->s1;
 free(p);
 p = l->s1;
 }else{
 if (!p) e = l;
 p = l;
 l = l->s1;
 }
 }
 return e;
}`

B) `elem *f(elem *l, int val){
 elem *p=0, *h=0;
 while (l){
 if (l->v == val) {
 if (p)
 p->s1 = l->s1;
 free(l);
 l = p->s1;
 }else{
 if (!p) h = l;
 p = l;
 l = l->s1;
 }
 }
 return h;
}`

(C) `elem *f(elem *l, int val){
 elem *p=0, *e, *h=0;
 while (l){
 if (l->v == val){
 e = l->s1;
 free(l);
 if (p) p->s1 = e;
 l = e;
 }else{
 if (!p) h = l;
 p = l;
 l = l->s1;
 }
 }
 return h;
}`