

### Treći deo ispita iz Programiranja 2

Trajanje: **105** minuta

#### Napomene:

- a) Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.  
 b) Vrednost odgovora: tačan = **15**; netačan = **-3.75**; nevažeći (nula ili više zacrtnjenih kružića) = **0**.  
 c) Na pitanjima se može osvojiti najviše **45** poena.  
 d) Prvi zadatak nosi **25**, a drugi **30** poena.

#### I. ZADACI

**1)** Napisati potprograme na jeziku C koji rade određenu obradu nad nizom znakova (stringom) koji sadrži pozitivan ceo broj u heksadecimalnom obliku. Potrebno je realizovati potprograme za unos, proveru ispravnosti i konverziju sadržanog heksadecimalnog broja u decimalni oblik. Smatrati da heksadecimalni broj nema više od 8 cifara i da može sadržati samo cifre od 0 do 9 i slova od A do F. Potprogram za proveru treba da provjeri da li uneseni string sadrži ispravan broj u heksadecimalnom obliku. Potprogram za konverziju heksadecimalnog broja u decimalni oblik ne sme da koristi bibliotečke funkcije. Napisati glavni program koji pozove potprogram za unos heksadecimalnog broja, provjeri ispravnost unesene stringa i, ukoliko je ispravan, ispiše njegovu decimalnu vrednost. Komunikacija između glavnog programa i funkcija, kao i između funkcija treba da se obavlja isključivo preko argumenata i povratnih vrednosti.

**2)** Na jednom SMS kvizu nagrade se dele po principu najmanje jedinstvene ponude – nagradu dobija onaj učesnik koji je ponudio najmanju jedinstvenu cenu za neku nagradu. Najmanja jedinstvena cena podrazumeva da je učesnik ponudio najmanju cenu koju niko drugi nije ponudio. U jednom redu tekst datoteke `ponude.txt` se nalaze podaci o poslatim ponudama po formatu `broj_telefona` (do 20 karaktera) `cena` (ceo broj). Jedan red tekst datoteke `ucesnici.txt` sadrži podatke o učesnicima kviza po formatu: `ime` (do 20 karaktera) `prezime` (do 20 karaktera) `broj_telefona` (do 20 karaktera). Podaci u obe datoteke su razdvojeni jednim znakom razmaka. Napisati program na programskom jeziku C koji pročita sadržaj navedenih tekst datoteka, odredi pobjednika SMS kviza i ispiše njegovo ime i prezime, broj telefona i ponuđenu cenu. U slučaju da nema pobjednika, program ispiše odgovarajuću poruku. Program treba da provjeri uspešnost poziva bibliotekskih funkcija i prijavi eventualne greške.

#### II. PITANJA

- 1)** Koja od sledećih tvrdjenja su tačna na programskom jeziku C?  
**(A)** Funkcija `malloc()` vraća nevalidan (NULL) pokazivač kao signal greške.  
**(B)** Definicija `char *(*f)(int *x)[ ]` je ispravna.  
 C) Veličina podatka tipa unije je jednaka zbiru veličina polja te unije.

**2)** Šta ispisuje sledeći program na programskom jeziku C ukoliko jednostruko ulančana lista sadrži redom brojeve 3 2 5 8 7? Smatrati da funkcija `ucitajlistu()` ispravno formira, a funkcija `pisilistu()` ispravno ispisuje jednostruko ulančanu listu.

|   |   |
|---|---|
| <pre>typedef struct elem {     int broj; struct elem *sled; } Elem; Elem *neznam(Elem *lst) {     Elem *p1=0, *p2=lst;     while(p2-&gt;sled!=0) {         p1=p2; p2=p2-&gt;sled;     }     p1-&gt;sled=lst;     p2-&gt;sled=lst-&gt;sled;     lst-&gt;sled=0;     return p2; }</pre> | <pre>void main () {     Elem *lst = ucitajlistu();     lst=neznam(lst);     pisilistu(lst); }</pre> |
|---|---|

A) 3 2 5 8 7

B) 7 8 5 2 3

**(C)** 7 2 5 8 3

**3)** Šta ispisuje sledeći program na programskom jeziku C?

|  |   |
|--|---|
| <pre>#include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; #include &lt;string.h&gt; void strnesto(char *a, int b) {     char *x = a;     while (*x) {         if (b&amp;&amp; *x&gt;='a'&amp;&amp;*x&lt;='z') {             *x+='A'-'a';         }         b&gt;&gt;=2;         x++;     } }</pre> | <pre>void main() {     char *str = malloc(12);     strcpy(str, "budi");     strnesto(str, 45);     strcpy(str, "pazljiv");     strnesto(str+5, 39);     printf("%s", str);     free(str); }</pre> |
|--|---|

A) BuDipAzLjiv

B) BUDiPAzLjiv

**(C)** bUDipAzLjiv

**4)** Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
int ff(int second, int first) {
    if (first == 0)
        return ff(second-1, 1);
    if (second == 0)
        return first+1;
    return ff(second-1, ff(second, first-1));
}
int main() {
    printf("%d %d %d", 1, 4, ff(1, 2));
    return 0;
}
```

A) 1 4 3

**(B)** 1 4 4

C) 1 4 6

## Treći kolokvijum iz Programiranja 2

Kolokvijum traje 105 minuta

### Napomene:

- Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.
- Vrednost odgovora: tačan = 15; netačan = -4; nevažeći (nula ili više zacrnjenih kružića) = 0.
- Na pitanjima se može osvojiti najviše 45 poena.
- Prvi zadatak nosi 30 poena, a drugi zadatak nosi 25 poena.

### I. ZADACI

1) U nekoj kompaniji, podaci o službenim automobilima se čuvaju u datoteci **automobili.txt**. Svaki red datoteke sadrži podatke o jednom automobilu – najpre registarski broj automobila (string od tačno 6 karaktera), a zatim naziv automobila koji se može sastojati od više reči. U datoteci **putovanja.txt** se nalaze podaci o službenim putovanjima – u jednom redu datoteke se nalazi datum (u formatu **dd-mm-gggg**), registarski broj korišćenog automobila i broj pređenih kilometara (ceo broj). Napisati program na programskom jeziku C koji pročita sadržaj navedenih tekst datoteka i ispiše na glavnom izlazu registarski broj i naziv automobila koji su prešli najmanji i najveći broj kilometara. U slučaju postojanja više automobila koji su prešli isti broj kilometara, uzima se u obzir onaj koji se pojavljuje prvi u datoteci.

2) Napisati program na programskom jeziku C koji u zadatom nizu znakova (stringu) ispravlja sve štamparske greške kod kojih prva dva znaka reči čine velika slova tako što drugi znak reči pretvori u malo slovo (npr. JUn treba da se zameni sa Jun). Reči su razdvojene pomoću jedne ili više razmaknica. Potrebno je realizovati funkcije za unos, obradu i ispis rezultujućeg stringa. Prilikom unosa, dužina stringa nije poznata unapred, već se čitanje vrši sve dok se ne unese znak za kraj reda. Napisati glavni program koji pozove funkcije za unos stringa, obradu i ispis rezultata. Program treba da ponavlja izvršavanje sve dok se sa glavnog ulaza ne unese prazan string. **Napomena:** komunikacija između glavnog programa i funkcija, kao i između samih funkcija treba da se obavlja isključivo preko argumenata i povratnih vrednosti. Voditi računa o ispravnoj alokaciji i dealokaciji dinamičke memorije.

### II. PITANJA

1) Koja od sledećih naredbi na jeziku C ispravno definiše pokazivač na funkciju koja prihvata argument tipa niz pokazivača na podatke tipa **double**, a vraća podatak tipa pokazivač na **char**?

- A) `char **x(double *y);`      B) `double *(*a)(char b[]);`      C) `char *(*p)(double *f[]);`

2) Šta ispisuje sledeći program na programskom jeziku C (pod pretpostavkom da dinamička alokacije memorije uvek uspeva)?

|   |  |
|---|--|
| <pre>#include &lt;stdio.h&gt; #include &lt;string.h&gt; void funkcija(int x, int b) {     if (x) {         funkcija(x/b,b);         b++; x++;     }     printf("%d",x%b); }</pre> | <pre>void main() {     int m,n;     m = strlen("Programiranje_2");     n = strlen("ETF");     funkcija(m,n); }</pre> |
|---|--|

(A) 0220

B) 0211

C) 1211

3)Šta radi sledeći fragment koda na programskom jeziku C, pod pretpostavkom da dinamička alokacija memorije uvek uspeva? Polja `prvi`, `posl` i `tek` strukture `Lista` pokazuju na prvi, poslednji i tekući element liste, respektivno. Smatrati da lista nije prazna i da `tek` nije `NULL`.

```
typedef struct elem { int broj; struct elem *pret, *sled; } Elem;
typedef struct { Elem *prvi, *posl, *tek; } Lista;
void dodaj (Lista *lst, int b) {
    Elem *novi = malloc (sizeof(Elem));
    novi->broj = b;
    novi->pret = lst->tek->pret;
    novi->sled = lst->tek;
    if (! lst->tek->pret) lst->prvi = novi;
    else lst->tek->pret->sled = novi;
    lst->tek = lst->tek->pret = novi;
}
```

(A) Dodaje element u dvostruko ulančanu listu ispred tekućeg elementa i dodati element postaje tekući.

B) Dodaje element u dvostruko ulančanu listu iza tekućeg elementa i dodati element postaje tekući.

C) Dodaje element u dvostruko ulančanu listu na početak liste i dodati element postaje tekući.

4)Šta ispisuje sledeći fragment koda na programskom jeziku C?

```
#include <stdio.h>
typedef int (*func)(int *);

int p1(int *a) {
    return *a++;
}
int p2(int *b) {
    return *b*2;
}
int p3(int *c) {
    return c[0]+c[1]+c[2];
}
```

```
main() {
    func nf[] = { p1, p2, p3 };
    func nf2[6] = { 0 };
    func *pf;
    int d[4][4] = { { 0, 1, 2 }, { 3, 4, 5 } };
    int i;

    for(i=0; i<3; i++) nf2[i<<1] = nf[i];
    for(pf=nf2+5,i=6;pf>=nf2;i--,pf--)
        if( *pf )
            printf("%d",(*pf)(&d[i%3][(i+2)%3]));
}
```

A) 042

B) 003

(C) 043

## Treći kolokvijum iz Programiranja 2

Kolokvijum traje 105 minuta

### Napomene:

- a) Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.  
 b) Vrednost odgovora: tačan = 15; netačan = -4; nevažeći (nula ili više zacrtnjenih kružića) = 0.  
 c) Na pitanjima se može osvojiti najviše 45 poena.  
 d) Prvi zadatak nosi 30 poena, a drugi zadatak nosi 25 poena.

### I. ZADACI

1) Neka se u datoteci `teniseri.txt` nalaze podaci o teniserima po sledećem formatu: šifra igrača (ceo broj), ime (najviše 30 znakova), prezime (najviše 30 znakova) i broj poena na ATP rang listi (ceo broj). U datoteci `wimbledon.txt` se nalaze podaci o plasmanu tenisera na ovom teniskom turniru. U svakom redu datoteke se nalazi šifra tenisera (ceo broj), broj poena koji brani na turniru (ceo broj) i broj poena koji je osvojio na turniru (ceo broj). Raspored tenisera u datotekama ne mora biti identičan, niti svi teniseri iz prve datoteke moraju postojati u drugoj datoteci. Napisati program na programskom jeziku C koji pročita sadržaj ulaznih tekst datoteka i formira jednostruko ulančanu listu, a zatim u izlaznu datoteku `atplista.txt` za svakog tenisera upiše novi broj poena po formatu kao u prvoj ulaznoj datoteci. Novi broj poena se dobija tako što se od starog broja poena na ATP listi oduzme broj poena koje teniser brani, a zatim doda broj poena koje je teniser osvojio na turniru. Voditi računa o ispravnom korišćenju zauzetih resursa.

2) Napisati program na programskom jeziku C koji prepisuje sadržaj ulazne tekst datoteke u izlaznu tekst datoteku. U izlaznu datoteku je potrebno prepisati sve one redove ulazne tekst datoteke koji predstavljaju palindrom (čitaju se isto i sleva na desno i sdesna na levo). Imena ulazne i izlazne datoteke se zadaju kao prvi i drugi argument komandne linije. Ukoliko se za ime izlazne datoteke zada string "-" sadržaj treba ispisati na standardni izlaz (`stdout`). Proveru da li određeni string predstavlja palindrom realizovati kao zaseban potprogram koji sa glavnim programom komunicira isključivo putem argumenata i povratnih vrednosti. Smatrati da jedan red ulazne datoteke ne sadrži više od 80 znakova.

### II. PITANJA

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

- A) Funkcija `fopen()` vraća vrednost `EOF` ukoliko ne može da otvori datoteku.  
 (B) Binarni operator `->` se koristi za pristup poljima strukture putem pokazivača.  
 C) Veličina unije je uvek jednaka zbiru veličina svih komponentata unije.

2) Šta ispisuje sledeći program na programskom jeziku C ukoliko jednostruko ulančana lista sadrži redom brojeve 3 2 5 8 7? Smatrati da funkcija `ucitajlistu()` ispravno formira, a funkcija `pisilistuu()` ispravno ispisuje sadržaj jednostruko ulančane liste.

|  |   |
|--|---|
| <pre>typedef struct elem {     int broj; struct elem *sled; } Elem; void obradi (Elem* lst) {     Elem *lst1 = lst, *lst2 = lst-&gt;sled;     if (!lst) return;     while(lst2) {         int t = lst1-&gt;broj;         lst1-&gt;broj = lst1-&gt;broj &gt; lst2-&gt;broj ? lst2-&gt;broj : t;         lst2-&gt;broj = t &gt; lst2-&gt;broj ? t : lst2-&gt;broj;;         if (!lst2-&gt;sled) break;         lst1 = lst2-&gt;sled;         lst2 = lst1-&gt;sled;     } }</pre> | <pre>void main (){     Elem *lst = ucitajlistu();     obradi(lst);     pisilistuu(lst); }</pre> |
|--|---|

(A) 2 3 5 8 7

(B) 7 8 5 2 3

(C) 2 3 8 5 7

3) Šta ispisuje sledeći program na programskom jeziku C ako se putem komandne linije proslede brojevi 3 4 7 5 3 5?

```
#include <stdlib.h>
#include <stdio.h>
void main (int argc, char *argv[]) {
    int i, b, a = atoi(argv[1]);
    do {
        b = atoi(argv[--argc]);
        if (b >= a) printf("%d ", a = b);
    } while (argc > 1);
}
```

A) 5 7

(B) 5 5 7

C) 4 7

4) Šta ispisuje sledeći program na programskom jeziku C?

|  |   |
|--|---|
| <pre>#include &lt;stdio.h&gt; int pp(int y, int *z, int a){     *z = y + a; a--;     return *z &gt; y ? pp(a, z, y) : a; }</pre> | <pre>void main(){     int x = 3, y = 1, z;     z = pp(x,&amp;y,y);     printf("%d %d %d", x, y, z); }</pre> |
|--|---|

A) 3 2 0

(B) 3 2 -1

C) 3 1 0

## Treći kolokvijum iz Programiranja 2

Kolokvijum traje 105 minuta

### Napomene:

- Pažljivo pročitajte Uputstvo pre popunjavanja Obrasca za odgovore.
- Vrednost odgovora: tačan = 15; netačan = -4; nevažeci (nula ili više zacrnjenih kružića) = 0.
- Na pitanjima se može osvojiti najviše 45 poena.
- Prvi zadatak nosi 30 poena, a drugi zadatak nosi 25 poena.

### I. ZADACI

1) Tekstualna datoteka `kalendar.txt` u svakom redu sadrži informacije o zakazanim aktivnostima u jednom danu. Za svaku aktivnost navodi se vreme početka, vreme završetka i kratak opis. Format jednog reda datoteke je dat u priloženom primeru. Napisati program na programskom jeziku C koji čita red po red iz datoteke, pravi strukturu aktivnosti na osnovu podataka iz pročitanih redova i stavlja je u listu u hronološkom poretku. Potom, polazeći od početka liste pronalazi aktivnosti koje se preklapaju. Ako se detektuje preklapanje između dve aktivnosti, iz liste se izbacuje aktivnost koja počinje kasnije ili, ako obe počinju u isto vreme, izbacuje se ona koja je druga u poretku. Aktivnost se može preklapati sa proizvoljno mnogo drugih aktivnosti. Aktivnosti koje se izbace iz liste, ispisuju se u izlaznu tekstualnu datoteku `preklapanja.txt`. Za kratak opis aktivnosti koristi se maksimalno 80 znakova. Na kraju program treba da ispravno oslobodi zauzete resurse. Učitavanje i brisanje liste realizovati kao potprograme.

Primer ulaza (`kalendar.txt`):

```
08:00-10:30 Ispit iz Osnova računarske tehnike.
10:00-10:30 Pregled kod doktora.
11:00-11:15 Preuzeti skripte iz štamparije.
09:00-09:30 Odbrana domaćeg iz Praktikumuma iz programiranja 2.
```

Primer izlaza (`preklapanja.txt`):

```
09:00-09:30 Odbrana domaćeg iz Praktikumuma iz programiranja 2.
10:00-10:30 Pregled kod doktora.
```

2) Napisati program na programskom jeziku C koji prepisuje sadržaj ulazne tekst datoteke u izlaznu tekst datoteku uz izostavljanje svih pojavljivanja HTML tagova. HTML tagovi su oblika `<tag>` ili `</tag>`, gde `tag` može biti bilo koji niz karaktera uključujući i blanko znake. Jedan HTML tag može početi u jednom, a završiti se u nekom drugom redu. Imena ulazne i izlazne datoteke se zadaju kao prvi i drugi argument komandne linije. Ukoliko se za ime izlazne datoteke zada string "-" sadržaj treba ispisati na standardni izlaz (`stdout`). Očuvati uređenost tekst po redovima.

### II. PITANJA

- Koja od sledećih tvrdjenja su tačna na programskom jeziku C?
  - Dozvoljeno je definisanje pokazivača na funkcije koje imaju povratni tip `void`.
  - Funkcija `fgetc()` vraća vrednost `EOF` ukoliko dođe do greške prilikom pristupa datoteci.
  - Definicija `int *p (int*, int)[10]` je dozvoljena.

2) Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
int f(int p) {
    if (p == 0) return 1;
    if (p == 1) return 2;
    return f(p-1)*f(p-2);
}

int main() {
    printf("%d %d", 4, f(4));
    return 0;
}
```

(A) 4 8

B) 4 6

C) 4 16

3) Šta ispisuje dati program, čija je izvršna verzija u datoteci `jun2012.exe`, ukoliko se pokreće preko komandne linije sledećom komandom (znak pitanja se ne prosleđuje): `jun2012.exe 4 3 2 5 ?`

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

(A) 36

B) 16

C) 10

4) Šta treba da stoji na mestu ##### da bi funkcija `ubaci` ispravno umetala novu vrednost u neopadajuće uređenu listu? Funkcija prihvata pokazivač na početak liste i vrednost koju treba umetnuti, a vraća pokazivač na listu nakon umetanja.

```
typedef struct elem {
    int broj;
    struct elem *sled;
} Elem;

Elem* ubaci(Elem *lst, int broj){
    Elem *tek=lst, *pret=NULL;
    while(tek)
        if (tek->broj < broj)
            pret=tek, tek=tek->sled;
        else break;
    Elem* novi=malloc(sizeof(Elem));
    novi->broj=broj;
    #####
    return lst;
}
```

(A)

```
if (!pret) {
    lst=novi; lst->sled=tek;
} else { pret->sled=novi;
    pret->sled->sled=tek;
}
```

B)

```
tek->sled=novi;
pret=novi;
if (tek==lst) lst=novi;
```

(C)

```
novi->sled=(!pret)?lst:tek;
if (!pret) lst=novi;
else pret->sled=novi;
```

## Treći kolokvijum iz Programiranja 2

Kolokvijum traje 105 minuta

### Napomene:

- a) Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.  
 b) Vrednost odgovora: tačan = 15; netačan = -3.75; nevažeci (nula ili više zacrnjenih kružića) = 0.  
 c) Na pitanjima se može osvojiti najviše 45 poena.  
 d) Prvi zadatak nosi 30 poena, a drugi zadatak nosi 25 poena.

### I. ZADACI

1) Napisati program na programskom jeziku C koji vrši određenu obradu nad tekst datotekom koja predstavlja plejlistu muzičkih numera. Svaki red ulazne i izlazne datoteke sadrži podatke o jednoj muzičkoj numeri po sledećem formatu: naziv numere (najviše 255 znakova i može sadržati blanko znake) i trajanje numere u formatu **mm:ss**, gde su **mm** minuti, a **ss** sekunde. Program treba da pročita ulaznu datoteku i formira jednostruko ulančanu listu muzičkih numera, a zatim učita jedan ceo broj **length** koji predstavlja dužinu nove plejliste u sekundama. Program treba da na osnovu učitanih podataka formira novu plejlistu čije ukupno trajanje neće biti duže od zadate dužine **length**. Novu plejlistu formirati tako da u nju stane što više numera. Imena ulazne i izlazne datoteke se zadaju kao prvi i drugi argument komandne linije. Voditi računa o korektnoj upotrebi zauzetih resursa.

| Primer ulazne datoteke                  | Primer izlazne datoteke za length = 570 s |
|---|---|
| Toto - Africa 04:56                     | Ray Charles - Hit the road Jack 01:59     |
| The Shadows - Apache 02:52              | The Shadows - Apache 02:52                |
| Eagles - Hotel California 06:30         | Whitesnake - Fool for your loving 04:17   |
| Whitesnake - Fool for your loving 04:17 |   |
| Ray Charles - Hit the road Jack 01:59   |   |

2) Napisati potprogram na programskom jeziku C koji pronalazi dve međusobno najudaljenije tačke u prostoru. Kao ulazni argument, potprogram dobija niz tačaka koji treba da obradi i njegovu dužinu, a vraća pozicije tačaka u nizu koje zadovoljavaju traženi uslov, kao i njihovu razdaljinu. Ukoliko postoji više tačaka koje zadovoljavaju traženi uslovi, vratiti prvi par na koji se naiđe. Svaka tačka je predstavljena realnim x, y i z koordinatama u Dekartovom koordinatnom sistemu. Napisati program na programskom jeziku C koji učita tačke sa standardnog ulaza u dinamički alocirani niz tačaka, pozove realizovani potprogram za računanje razdaljine i ispiše dobijene rezultate. Potprogram sa glavnim programom treba da komunicira isključivo pomoću argumenata i povratne vrednosti.

### II. PITANJA

1) Šta ispisuje sledeći program napisan na programskom jeziku C ukoliko se putem komandne linije pozove kao **pitalica 4 23 17**?

|  |  |
|--|--|
| <pre>int fir(int k, int n) {     if (k &gt;= n) return 0;     else if ((n % k) == 0) return 1;     else return fir(k + 1, n); } int ip(int n) {     return ((n &gt; 1) &amp;&amp; !fir(2, n)); }</pre> | <pre>#include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; void main (int argc, char* argv[]) {     int i;     for (i = 1; i &lt; argc; i++)         printf("%d", ip(atoi(argv[i]))); }</pre> |
|--|--|

A) 111

(B) 011

C) 001

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

- A) Pokazivači na funkcije se mogu definisati samo za funkcije koje nemaju formalne argumente.  
 (B) Ako su **s1** i **s2** strukturne promenljive istog tipa, dozvoljena je dodela vrednosti **s1 = s2**;  
 C) Element jednostruko ulančane liste obavezno sadrži pokazivač na prethodni i pokazivač na sledeći element u listi.

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

|   |  |
|---|--|
| <pre>#include &lt;stdio.h&gt; #include &lt;stdlib.h&gt;  typedef struct { int x, y; } Data;  int f1 (Data *pd, Data d);  void main() {     Data d;     d.x = 4, d.y = 2;     printf("%d", f1(&amp;d, d)); }</pre> | <pre>int f1 (Data *pd, Data d) {     int result;     if ((pd-&gt;x + pd-&gt;y) &gt; 0) {         Data d1;         d1.x = pd-&gt;x-2;         d1.y = pd-&gt;y++;         result = f1(&amp;d1, *pd);         return result + d.y;     }     else         return d.x+d.y; }</pre> |
|---|--|

A) 13

(B) 9

C) 15

4) Šta ispisuje sledeći fragment koda na programskom jeziku C?

|  |   |
|--|---|
| <pre>#include &lt;stdio.h&gt;  typedef void (*FPtr)(int*); typedef void (*FA)(FPtr*, int*);  void m1(int *a) { *a++; } void m2(int *b) { *b+=2; } void m3(int *c) { (*c)++; }  void x(FPtr *farr, int* a) {     (++farr)(a);     (*farr)(a+4); }</pre> | <pre>void y(FPtr *farr, int *a) { (*(farr+1))(a); }  int main() {     int n[5] = {1, 4, 7}, i;     FPtr a[] = { m1, m3, m2 };     FA b[] = { x, y };     for (i=0; i&lt;2; i++) {         b[i](a+i, n+i);     }     for (i=0; i&lt;5; i++) printf("%d ", n[i]);     return 0; }</pre> |
|--|---|

A) 2 6 7

B) 2 6 8 0 0

(C) 2 6 7 0 1

## Treći kolokvijum iz Programiranja 2

Kolokvijum traje **105** minuta

### Napomene:

- a) Pažljivo proučite Uputstvo pre popunjavanja Obrasca za odgovore.  
 b) Vrednost odgovora: tačan = **15**; netačan = **-3.75**; nevažeći (nula ili više zacrnjenih kružića) = **0**.  
 c) Na pitanjima se može osvojiti najviše **45** poena.  
 d) Prvi zadatak nosi **30** poena, a drugi zadatak nosi **25** poena.

### I ZADACI

1) U datoteci *prodavnica.txt* nalazi se spisak namirnica koje se mogu kupiti u nekoj prodavnici. U svakom redu se nalazi naziv namirnice (reč do 30 znakova), količina (ceo broj) i jedinična cena namirnice (realan broj). U datoteci *potrepstine.txt* nalazi se spisak namirnica koje treba kupiti. U svakom redu datoteke nalazi se naziv namirnice (reč do 30 znakova) i potrebna količina (ceo broj). Napisati glavni program na programskom jeziku C koji na osnovu opisanih ulaznih datoteka formira izlaznu datoteku *faktura.txt* u kojoj se nalaze potrebne namirnice koje je moguće kupiti u prodavnici. U svakom redu izlazne datoteke nalazi se naziv namirnice, količina koju je moguće kupiti u prodavnici, jedinična cena namirnice, kao i ukupna cena namirnice. Na kraju datoteke treba da se nalazi i ukupna cena svih namirnica u fakturi. Ukoliko se neka namirnica iz spiska potrepština uopšte ne prodaje u prodavnici ili je nema u dovoljnoj količini, ona se ne ubacuje u fakturu. Voditi računa o pravilnom rukovanju korišćenim resursima. Sadržaj ulaznih datoteka je dozvoljeno pročitati samo jednom, a maksimalan broj redova u datotekama, kao ni raspored stavki po redovima, nije u napred poznat. Primer datoteka:

| prodavnica.txt | potrepstine.txt | faktura.txt          |
|----------------|-----------------|----------------------|
| kupus 100 30.5 | pivo 20         | pivo 20 62.5 1250.00 |
| ulje 15 122.8  | jabuke 30       | 1250.00              |
| pivo 50 62.5   | ulje 25         |                      |

2) Napisati potprogram na programskom jeziku C koji vrši obradu nad kvadratnom matricom celih brojeva, tako što u svaki element matrice ispod sporedne dijagonale čija je vrednost prost broj upiše vrednost njemu simetričnog elementa u odnosu na sporednu dijagonalu. Potprogram kao argumente dobija pokazivač na dinamičku matricu i dimenziju matrice. Napisati glavni program kojem se dimenzija matrice prosleđuje kao argument komandne linije. Program na osnovu zadate dimenzije napravi dinamičku kvadratnu matricu i incijalizuje elemente vrednostima unesenim sa standardnog ulaza. Program potom pozove potprogram za obradu matrice i ispiše elemente matrice nakon obrade. Voditi računa o pravilnom rukovanju korišćenim resursima.

### II PITANJA

1) Koje od ponuđenih tvrdnji su tačne u skladu sa standardom programskog jezika C?

- A) `int (*p) [1] (int *a);` je potpuno ispravna definicija promenljive `p`.  
 B) Ako su date definicije: `struct flasa {char boja[10]; float zapremina;}a;` `struct boca {char boja[10]; float zapremina;}b;` tada je dozvoljena dodela `a=b;`.  
 (C) Upisom u jedno polje unije moguće je promeniti vrednost drugog polja te unije.

2) Šta ispisuje sledeći program na programskom jeziku C, ako je sadržaj argumenata komandne linije: `nenenedadada 3 x x x x?`

|  |  |
|--|--|
| <pre>#include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; char* rpc(int num, char* str, char ch, char tr){     int i=0;     for(; *str &amp;&amp; i&lt;num; str++, i++);     for(; *str &amp;&amp; *str!=ch; str++);     if(*str) *str = tr;     return (*str) ? str : NULL; }</pre> | <pre>void main(int argc, char** argv){ char* str=argv[1], ch = *argv[1]; int num=atoi(argv[2]), rem=argc-2; while(--rem){ char* res = rpc(num, str, ch, argv[argc-rem][0]); if(res) putchar(*res); } printf(" %s\n", str); }</pre> |
|--|--|

- A) `xx nenexexedada`      B) `xxxx xexexexedada`      C) `xx xexenenedada`

3) Šta ispisuje sledeći program na programskom jeziku C?

|   |   |
|---|---|
| <pre>#include &lt;stdio.h&gt; #define X(a) sizeof(a)/sizeof(int) typedef int (*FuncPtr)(int*);  int f1(int *p) { return *p++; } int f2(int *q) { return *(q+1)++ + *q; } int f3(int *r) { return ++r[-2]; }</pre> | <pre>void main() { FuncPtr f[] = {f1, f3, f2, f1}; int a[] = {2, 3, 5, 7, 9}, i = X(a); while (--i &gt; 0) a[i-1] = f[i-1](a+i); for (i=0; i&lt;5; printf("%d", a[i++])); }</pre> |
|---|---|

- A) 3210980      (B) 3318910      C) 3259710

4) Šta ispisuje sledeći program na programskom jeziku C?

|   |  |
|---|--|
| <pre>#include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; typedef struct {int a, b;} S1; typedef struct { S1* ps1; int c; union {int* d; int* e;} u; } S2; void main() { S1 s1; S2 s2, s22; int a = 6, b = 8; }</pre> | <pre>s1.a = 1; s1.b = 2; s2.ps1 = &amp;s1; s2.c = 3; s2.u.d = &amp;a; s2.u.e = &amp;b; s22 = s2; s2.ps1-&gt;b = 5; *s2.u.e = 10; printf("%d %d %d %d", s22.ps1-&gt;a, s22.ps1-&gt;b, s22.c, *s22.u.d, *s22.u.e);</pre> |
|---|--|

- A) 1 2 3 6 8      (B) 1 5 3 10 10      C) 1 5 3 6 10