

## 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; nevažeci (nula ili više zacrtnjenih kružića) = 0.  
 c) Na pitanjima se može osvojiti najviše 25 poena.  
 d) Zadaci nose po 20 poena.

### I. ZADACI

1) Neka se u datoteci **meteo.txt** nalaze podaci o vremenskim prilikama na određenim meteorološkim mernim mestima. U svakom redu se nalazi naziv mernog mesta (niz od najviše 30 znakova), temperatura (realan broj), pritisak (realan broj), brzina vetra (realan broj) i vlažnost vazduha (realan broj). Napisati program na programskom jeziku C koji pročita sadržaj ulazne tekst datoteke i formira jednostruko ulančanu listu, a zatim na standardni izlaz ispiše nazive, temperaturu i vlažnost vazduha na svim onim mernim mestima kod kojih je izmeren vazdušni pritisak između 1010 i 1015 milibara, a brzina vetra je manja od 1 m/s. Voditi računa o ispravnom korišćenju zauzetih resursa.

2) Napisati program na programskom jeziku C za obradu i kriptovanje poruka. Poruka se unosi sa standardnog ulaza sve dok se ne unese znak za novi red. Poruku sačinjavaju mala i velika slova engleske abecede, zapete, razmaci i tačke. Maksimalna dužina poruke je 80 znakova. Nakon unosa i smeštanja poruke u niz, sva višestruka, uzastopna pojavljivanja blanko znaka zameniti jednim blanko znakom, a zatim izvršiti kriptovanje. Kriptovanje pripremljene poruke se vrši tako što se redom za svaki znak u poruci generiše pseudoslučajan broj u opsegu od 0 do 255, a zatim se na tekući karakter i generisani psedoslučajni broj primeni bitska operacija „ekskluzivno ili“ i dobijena vrednost smesti u novi niz. Elemente niza ispisati na standardnom izlazu u heksadecimalnom brojnem sistemu. Učitavanje, obradu (uklanjanje suvišnih razmaka i kriptovanje) i ispis rezultata implementirati kao zasebne potprograme koji sa glavnim programom komuniciraju isključivo putem liste argumenata i povratne vrednosti.

### II. PITANJA

1) Šta ispisuje sledeći program na programskom jeziku C ako se putem komandne linije proslede brojevi 2 7 4 5 1 3?

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

- (A) 3 4 3 5                      (B) 3 4 7                      (C) 4 5 2 7

2) Šta ispisuje sledeći program na programskom jeziku C?

<pre>#include &lt;stdio.h&gt; int x = 3, y = 1, z = 2; int r(int x, int *y){     z = x + *y;     if (z&gt;7) return *y+x;     else {         z = r(z, y); return z -= x;     } }</pre>	<pre>void main(){     z = r(z,&amp;z);     printf("%d", x + y + z); }</pre>
--	---

- (A) 9                      (B) 14                      (C) 12

3) Neka je **void obrni(char \*s);** funkcija koja obrće string na koji pokazuje pokazivač p. Koja realizacija ove operacije je ispravna? Pretpostaviti da je string nenulte dužine.

- |   |   |   |
|---|---|---|
| (A)   | (B)   | (C)   |
| <pre>char c,*p = s; while(++s); while(--s&gt;p){     c=*s;*s=*p;*p++=c; }</pre> | <pre>char c,*p = s; while(++s); while(s--&gt;p++){     c=*s;*s=*p;*p=c; }</pre> | <pre>char c,*p = s; while(++s); while(s&gt;p){     c=*s;*s--=*p;*p++=c; }</pre> |

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

- |                       |                          |                           |
|-----------------------|--------------------------|---------------------------|
| (A)                   | (B)                      | (C)                       |
| int **(*x(int *))( ); | char *(*x(int [ ])) [ ]; | char *(*(*x)(int *)) [ ]; |

5) Šta ispisuje sledeći program na programskom jeziku C?

<pre>#include &lt;stdio.h&gt; int f1 (int *a) {     return ++*a; } int f2 (int *a) {     return a[-1]+a[0]+a[1]; }</pre>	<pre>void main ( ) {     int a[4][4] = {{3,2,1},{4,5},{6,5,8,7}}; int i;     for (i=2;i&gt;0;i--) {         printf("%d %d ", f1(a[3-i]+i), f2(&amp;a[i][3-i]));     } }</pre>
--	---

- (A) 7 19 5 5                      (B) 1 19 6 6                      (C) 1 20 6 5

6) 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 sadržaj jednostruko ulančane liste.

<pre>typedef struct elem {     int broj; struct elem *sled; } Elem; Elem *f(Elem *h){     Elem *p = h,*e = h;     while (e-&gt;sled){         p = p-&gt;sled;         e = e-&gt;sled-&gt;sled;     }     return p; }</pre>	<pre>void main () {     Elem *lst1 = ucitajlistu();     Elem *lst2 = f(f(lst1));     pisilistu(lst2); }</pre>
--	---

- (A) 8 7                      (B) 5 7 8                      (C) 3 2 5