

**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 zacrtnjenih kružića) = **0**.  
 c) Na pitanjima se može osvojiti najviše **25** poena.  
 d) Zadaci nose po **20** poena.

**I. ZADACI**

**1)** Napisati funkciju `char* uporedi(char *str, char *podstr)` na programskom jeziku C koja proverava da li se u stringu `str` nalazi podstring zadat argumentom `podstr`. Ukoliko podstring postoji u stringu zatom kao prvi argument, funkcija treba da vrati pokazivač na prvi znak pronađenog podstringa, a u suprotnom vrednost NULL. Smatrati da podstring koji se traži mora biti u celosti u jednom redu i da ne sme da sadrži blanko znak ili tab. Napisati program na programskom jeziku C koji sa glavnog ulaza čita jedan red teksta i koristeći realizovanu funkciju ispisuje koliko se puta pojavio zadati podstring u učitanoj tekstu. Podstring koji se traži se navodi kao prvi argument komandne linije i ne sadrži blanko znak ili tab. Komunikaciju između glavnog programa i funkcije obavljati isključivo putem argumenta i povratne vrednosti.

**2)** Napisati program na programskom jeziku C koji proverava da li je od delova u magacinu moguće sklopiti traženi proizvod. Podaci o delovima u magacinu su zapisani u tekstualnoj datoteci "magacin.txt". U svakom redu su podaci o tačno jednom delu i to u sledećem formatu: šifra (ceo broj kojim se identifikuje deo) i raspoloživa količina (ceo broj). Opis delova potrebnih za sklapanje jednog traženog proizvoda je zapisan u datoteci "proizvod.txt". U svakom redu su zapisana dva cela broja: šifra dela i potrebna količina. Skup šifara koje se pojavljuju u datoteci "proizvod.txt" je podskup šifara koje se pojavljuju u datoteci "magacin.txt". Dozvoljeno je samo jednom pročitati obe datoteke. Na standardnom izlazu ispisati odgovarajuću od sledeće dve poruke poruku: "Moguće je sklopiti proizvod." ili "Nije moguće sklopiti proizvod."

**II. PITANJA**

**1)** Ako postoji deklaracija: `char *s;` koji od sledećih izraza na mestu `###` u naredbi `for(*s=1;*s###);` obezbeđuje pravilno učitavanje znakovnog niza (stringa), počevši od lokacije na koju ukazuje `s`? Smatrati da je unapred alocirano dovoljno memorije za smeštanje stringa.

- (A) `((s = getchar()) == '\n') ? (*s = 0) : (++s=1)`  
 B) `(s = getchar()), ((s == '\n') ? s = 0 : s++)`  
 C) `(*s++ = getchar()), (( *--s == '\n') ? *s = 0 : 0)`

**2)** Šta radi funkcija `obradi()`, ako joj se prosledi pokazivač na početak liste celih brojeva?

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

- A) Obrće poredak brojeva u ulančanoj listi na parnim pozicijama u listi.  
 B) Obrće poredak brojeva u ulančanoj listi na neparnim pozicijama u listi.  
 (C) Zamenjuje mesta brojeva u ulančanoj listi na sledeći način: prvi sa drugim, treći sa četvrtim itd.

**3)** Šta ispisuje sledeći program na programskom jeziku C? Napomena: pretpostaviti da ne dolazi do greške prilikom alokacije memorije.

<pre>#include &lt;stdio.h&gt; #include &lt;stdlib.h&gt; #include &lt;string.h&gt; int (**niz)(int); int broj[5] = { 0, 2, 1}; int f1(int n) {     printf("%d", broj[(n+5)%5]); } return (*niz[broj[(n+5)%5]])(n-1); else return 1;</pre>	<pre>int f2(int n) {     printf("%d", broj[(n+5)%5]);     if (n&gt;0) return (*niz[broj[(n+5)%5]])(n-2);     else return 2; } void main() {     niz=(int(**)(int))calloc(3,sizeof(int (*)(int)));     niz[0] = f1; niz[1] = f2; niz[2] = f1;     printf("%d\n",f1(4));     free(niz); }</pre>
--	---

A) 001202

B) 102102

(C) 001201

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

(A) Operator `<<` vrši obradu nad binarnom predstavom operanda na isti način bez obzira da li je prvi operand označen ili neoznačen broj.

B) Funkcija `main()` može da prihvati proizvoljan broj argumenata.

C) Naredba `continue` se unutar `switch` naredbe ponaša identično kao naredba `break`.

**5)** Šta će ispisati sledeći program na programskom jeziku C ako se program nalazi u fajlu `prog.exe` i ako mu se prilikom pokretanja proslede sledeći parametri: `bcd abcd ab abc`?

<pre>#include&lt;stdio.h&gt; void main(int argc, char* argv){     char *p, *q;     int i = 2, j = 0, k = 0;     for (;i &lt; argc; i++){         p = argv[i-1]; q = argv[i];</pre>	<pre>for(j = 0; *q &amp;&amp; *p; j++, p++, q++)     if (*p != *q) break;     if (!*p    !*q) k += i; } printf("%d",k); }</pre>
--	---

A) 9

B) 5

(C) 7

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

<pre>#include &lt;stdio.h&gt; int f(int *a, int b){     if (*a &gt; 0) return b + f(&amp;b, *a - 1);     else return *a; }</pre>	<pre>void main(){     int x;     x = 5;     printf("%d",f(&amp;x, 1)); }</pre>
--	--

A) 10

(B) 5

C) 7