

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 funkciju na programskom jeziku C koja prihvata najviše 300 zapisa o tačkama, gde svaki zapis sadrži dva cela broja, x i y koordinate tačke, i pronalazi pravougaonik čija su dva dijagonalna temena neke dve tačke iz zadatog niza i čije su stranice paralelne sa koordinatnim osama, a koji obuhvata najveći broj preostalih zadatah tačaka. Tačke koje se nalaze na stranicama se ne smatraju obuhvaćenim. Napisati program na programskom jeziku C koji sa standardnog ulaza čita podatke o najviše 300 tačaka, pronalazi traženi pravougaonik pozivom prethodno implementirane funkcije i ispisuje koordinate tačaka koje predstavljaju dijagonalna temena nađenog pravougaonika.

2) Napisati program na programskom jeziku C koji pomaže prilikom pakovanja proizvoda za transport. Podaci o proizvodima koje treba transportovati su dati u tekstualnoj datoteci "posiljka.txt". Podaci su zapisani tako da su u svakom redu podaci o tačno jednom proizvodu za slanje i to u sledećem formatu: šifra (ceo broj), zapremina potrebna za pakovanje (ceo broj, izražen u cm³) i naziv proizvoda (tekst, do 50 znakova). Proizvodi se pakuju u proizvoljan broj paketa, svaki zapremine 10000 cm³, po sledećem algoritmu. Uzima se jedan prazan paket i potom se taj paket puni sve dok ima proizvoda koji mogu da stanu u isti paket. Paket se puni tako što se bira najveći od preostalih proizvoda koji može da stane u paket koji se puni i taj proizvod se ubacuje u paket. Kada nijedan od preostalih proizvoda ne može da stane u paket koji se trenutno puni, prelazi se na novi prazan paket. Prethodni postupci se ponavljaju sve dok se ne upakuju svi proizvodi. Poznato je da svaki proizvod može stati u prazan paket. Kao rezultat rada programa, na standardni izlaz ispisati broj upotrebljenih paketa, kao i sadržaj svakog paketa. Za čuvanje podataka o proizvodima u memoriji koristiti jednostruko ulančanu listu sortiranu po nerastućoj vrednosti zapremine.

II. PITANJA

1) Ako postoji deklaracija: `char *s, *t;` koji od ponuđenih segmenata koda na ispravan način nadovezuje sadržaj stringa na koji pokazuje t na string na koji pokazuje s? Smatrati da je unapred alocirano dovoljno memorije za smeštanje stringa.

- | | | |
|---|---|--|
| A) <code>while (*s++);</code>
<code>while (*s++==*t++);</code> | B) <code>while (*s)</code>
<code>s++;</code>
<code>while (*s++==*t++, *t);</code> | (C) <code>while (*s++);</code>
<code>while (--s,*s++==*t++)</code>
<code>s++;</code> |
|---|---|--|

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, *tmp; if (!lst) return; lst2 = lst->sled; if (!lst2) return;</pre>	<pre>while(lst2->sled) { if (lst1->broj = lst2->sled->broj){ tmp = lst2->sled; lst2->sled = lst2->sled->sled; free(tmp); lst2->broj = lst1->broj; } lst1 = lst2; lst2 = lst1->sled; }</pre>
--	--

- (A) Izbacuje iz liste sve vrednosti kojima prethodi i sledi ista vrednost.
 B) Sva uzastopna pojavljivanja jedne vrednosti menja jednim pojavljivanjem te vrednosti.
 C) Od svake tri uzastopne vrednosti gde su prva i treća iste, izbacuje trecu vrednost.

3) Ako su znakovni nizovi ("stringovi") niz1 i niz2 respektivno "ISPIT" i "XYZ", funkcija uradi vraća pokazivač na znakovni niz:

```
#include <stdlib.h>
#include <string.h>

char *uradi( char *niz1, char *niz2 ){
    char *rez = malloc( 2 * strlen ( niz1 ) + 1 );
    char *pom = rez;
    while ( *pom++ = *niz2++ ) *pom++ = *niz1++;
    return rez;
}
```

- (A) XIYSZP B) ISPXYZ C) XYZISP

4) Šta ispisuje sledeći segment programa na programskom jeziku C?

```
#include <stdio.h>
main ( ) {
    int a[3][3] = { {1,2,3}, {4,5,6}, {7,8,9}}; int i;
    for (i=2;i>=0;i--){
        printf("%d %d %d \n",*(a[i]+1), a[2-i][i], *((a+i)+2-i));
    }
```

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

5) Šta će ispisati sledeći program na programskom jeziku C ako mu se prilikom pokretanja proslede sledeći parametri: 2 4 1 0 5 9 10 7 6 5?

<pre>#include<stdio.h> void main(int argc, char* argv[]){ int i = 2, a, b, j = -1, k = 0; a = atoi(argv[1]); for (;i < argc; i++){ b = atoi(argv[i]); if (j*a < j*b){</pre>	<pre> k++; j *= -1; } a = b; } printf("%d",k); }</pre>
---	---

- A) 7 B) 2 (C) 3

6) Šta ispisuje sledeći program na programskom jeziku C?

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

- A) 10 (B) 6 C) 8