

Ispit iz Programiranja 2

Ispit traje 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) Zadatak nosi 20 poena.

I ZADACI

1) 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 učitava 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.

2) 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 učitava 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	

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 >= n) return 0; else if ((n % k) == 0) return 1; else return fir(k + 1, n); } int ip(int n) { return ((n > 1) && !fir(2, n)); }</pre>	<pre>#include <stdio.h> #include <stdlib.h> void main (int argc, char* argv[]) { int i; for (i = 1; i < 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) Koja od sledeće tri funkcije na ispravan način vraća pokazivač na poslednju pojavu karaktera **c** u stringu **s** ili vrednost **NULL** ukoliko takav karakter ne postoji u stringu **s**?

<pre>char* strrchr1 (char *s, char c) { while (*s != '\0' && *s != c) s++; return ((*s == c) ? s : NULL); }</pre>	<pre>char *strrchr2 (char *s, char c) { char* last = NULL; while (*s) { last = *s == c ? s : NULL; s++; } return last; }</pre>	<pre>char *strrchr3 (char *s, char c) { char* last = NULL; do { if(*s == c) last = s; } while(*s++); return last; }</pre>
--	--	---

A) strrchr1

B) strrchr2

(C) strrchr3

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

<pre>#include <stdio.h> #include <stdlib.h> 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(&d, d)); }</pre>	<pre>int f1 (Data *pd, Data d) { int result; if ((pd->x + pd->y) > 0) { Data d1; d1.x = pd->x -- 2; d1.y = pd->y++; result = f1(&d1, *pd); return result + d.y; } else return d.x+d.y; }</pre>
--	---

A) 13

(B) 9

C) 15

5) Šta treba da stoji na mestu ##### da bi funkcija **remove** ispravno izbacivala zadati element iz dvostruko ulančane liste? Funkcija prihvata pokazivač na početak liste i pokazivač na element koji je potrebno izbaciti, a vraća pokazivač na početak liste nakon izbacivanja. Izbačeni element se ne dealocira.

<pre>typedef struct elem { int broj; struct elem *next, *prev; } Elem; Elem* removeElem (Elem *lst, Elem *e) { ##### return lst; }</pre>
--

(A)

B)

C)

<pre>if (e->next) e->next->prev = e->prev; if (e->prev) e->prev->next = e->next; if (e == lst) lst = e->next;</pre>	<pre>e->next->prev = e->prev; e->prev->next = e->next; if (e == lst) lst = e->next;</pre>	<pre>if (e->next) e->prev->next = e->next; if (e->prev) e->next->prev = e->prev; if (e == lst) lst = e->next;</pre>
--	--	--

6) Šta ispisuje sledeći program na programskom jeziku C?

<pre>int shuffle (int *a) { int t = a[-1], i, *p; for(p = a; p < a + 2; p++) *(p-1) = *p; p[0] = t; return *p; }</pre>	<pre>#include <stdio.h> void main () { int a[] = {3, 2, 1, 4, 5, 6, 9, 8}, i; for (i = 1; i < sizeof(a) / sizeof(int) - 1; i += 2) printf("%d ", shuffle(a + i)); }</pre>
---	--

A) 2 1 5

(B) 3 1 5

C) 3 1 5 9