

## Ispit iz Programiranja 2

Ispit traje 120 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 15 poena. Prvi zadatak nosi 25 poena, dok drugi nosi 30 poena.

### I ZADACI

1) Napisati program na programskom jeziku C koji vrši nadovezivanje ulaznih tekstualnih datoteka i rezultat upisuje u izlaznu tekst datoteku. Sadržaj datoteke čine slova i beli znakovi (razmaci i znakovi za novi red). Ime izlazne datoteke se zadaje kao prvi argument komandne linije, dok se imena ulaznih datoteka zadaju kao preostali argumenti komandne linije. Ulaznih datoteka može biti proizvoljno mnogo. Prilikom nadovezivanja teksta potrebno je izostaviti sve prazne redove. Na kraju izlazne datoteke ispisati ukupan broj slova, belih znakova, linija teksta i ukupan broj prepisanih reči u izlaznu datoteku. Dužina jednog reda u ulaznim datotekama nije ograničena. Reč predstavlja neprekidan niz slova.

2) Napisati program na programskom jeziku C koji u sekvenci pozitivnih celih brojeva pronalazi broj koji se najčešće pojavljuje. Program čita cele brojeve sa standardnog ulaza i ubacuje ih u jednostruko ulančanu listu pojavljivanja. Element liste pojavljivanja (*Occurrence*) sadrži vrednost učitano broja, brojač pojavljivanja i pokazivač na sledeći element. Za ubacivanje broja u listu pojavljivanja potrebno je napisati potprogram `void addOccurrence(Occurrence **list, int num)`. Potprogram prvo proverava da li broj postoji u listi. Ako postoji, ažurira mu se postojeći element. U suprotnom, na početak liste se dodaje novi element. Unos brojeva se završava kada korisnik unese nulu. Program treba da na standardni izlaz ispiše broj koji se najčešće pojavljavao na ulazu i njegov broj pojavljivanja. Ukoliko ima više takvih brojeva, ispisati ih sve. Voditi računa o korektnoj upotrebi korišćenih resursa.

### II PITANJA

1) Šta ispisuje sledeći program na programskom jeziku C ukoliko jednostruko ulančana lista sadrži redom brojeve 2 3 1 5 7 4 9? Smatrati da funkcija `loadList` ispravno formira, a funkcija `printList` ispravno ispisuje sadržaj liste redom od početka.

```
#include <stdio.h>
#include <stdlib.h>
typedef struct node {
    int data;
    struct node* next;
} Elem;
Elem* loadList(void);
void printList(Elem*);
void processList(Elem**, int);
void main () {
    Elem *lst = loadList();
    processList(&lst, 4);
    printList(lst);
}

void processList(Elem **head, int k)
{
    if (k == 0) return;
    Elem* curr = *head; int count = 1;
    while (count < k && curr != NULL) {
        curr = curr->next;
        count++;
    }
    if (curr == NULL) return;
    Elem *node = curr;
    while (curr->next != NULL) curr=curr->next;
    curr->next = *head; *head = node->next;
    node->next = NULL;
}
```

(A) 7 4 9 2 3 1 5

B) 1 3 2 9 4 7 5

C) 1 5 7 4 9 2 3

2) Šta treba da stoji umesto ##### i &&&& da bi sledeća funkcija na programskom jeziku C vraćala početnu poziciju poslednjeg pojavljivanja stringa `pat` u stringu `txt`? Funkcija vraća -1 ukoliko se string `pat` ne sadrži u stringu `txt`.

```
int s(char *pat, char *txt) {
    int i, j, m, n, r = -1;
    m = strlen(pat); n = strlen(txt);
    for (i = 0; i <= n - m; i++) {
        for (j = 0; j < m; j++)
            #####
            &&&&
        }
    return r;
}
```

(A)	B)	C)
####:	####:	####:
if (txt[i+j] != pat[j])	if (txt[i+j] == pat[j])	if (txt[i+j] != pat[j])
break;	continue;	break;
&&&&:	&&&&:	&&&&:
if (j == m) r = i;	if (i == m) r = j; continue;	if (i == m) r = j; break;

3) Šta će sledeća funkcija, napisana na programskom jeziku C, uraditi sa sadržajem niza `a` dužine `n`?

```
void f(int a[], int n){
    int x=a[n-1];
    if (n>1){
        a[n-1] = a[0];
        f(a+1, n-2);
        a[0] = x;
    }
}
```

- A) Iz niza će izbaciti svaki drugi element.  
 B) Elementi niza će biti obrnuti dva puta (ostaće u početnom poretku).  
 C) Redosled elemenata niza će biti obrnut.

4) Šta ispisuje sledeći program na programskom jeziku C? Program se nalazi u datoteci `obrada.exe` i pokreće se sledećom komandom: `obrada bundeva dubrava dabar 1 2 3`

```
#include <stdio.h>
#include <stdlib.h>
void main(int n, char* a[]) {
    int i, x = atoi(a[n-1]);
    char *t = a[x];
    x = atoi(a[n-1-x]);
    for(i = 0; a[x][i]; i++) {
        int j = 0;
        for(j = 0; *(t) != 0; j++) {
            if (a[x][i] == *t++)
                printf("%c", *t?*_t: '_');
        }
        t -= j;
    }
}
```

A) rav\_uv\_

B) a\_brabr

C) aa\_