

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 pomaže sastavljaču rasporeda časova. Za svaku salu sa standardnog ulaza se unosi najpre potreban broj sati u slobodnoj sali i broj studenata, a zatim se u proizvoljnom broju redova u svakom redu unosi broj sale, maksimalni broj studenata koji mogu da stanu u tu salu i niz nepreklopljenih, rastuće uređenih intervala u kojima je sala slobodna, po celobrojnim satima razdvojenim znakom -. Sati su u opsegu radnog vremena od 8h do 22h. Potrebno je pronaći sve sale koje su slobodne dovoljno dugo u kontinuitetu, i koje su dovoljno velike za zadati broj studenata. Ukoliko postoji više termina dovoljne dužine u istoj sali, bira se najraniji od njih. Ukoliko postoji više sala koje zadovoljavaju postavljene kriterijume, ispisati sve zadovoljavajuće sale uređene po neopadajućem kapacitetu sale. Na standardnom izlazu se za svaku salu ispisuje broj sale, kapacitet i odabrani vremenski interval. Ukoliko salu nije moguće naći, ispisati odgovarajuću poruku. U jednostruko ulančanu listu smeštati samo najraniji slobodan interval u sali odgovarajućeg kapaciteta. Element sadrži broj i kapacitet sale, kao i početno i krajnje vreme slobodnog intervala. Voditi računa o pravilnom rukovanju korišćenim resursima.

Ulaz	Izlaz	Ulaz	Izlaz
2 110	65 120 8-10	3 120	Ne postoji slobodan termin!
56 150 13-15 20-22	56 150 13-15	56 150 13-15 20-22	
65 120 8-10 12-14 17-18		65 120 8-10 12-14 17-18	
70 100 8-11 14-17 19-20 21-22		70 100 8-11 14-17 19-20 21-22	

2) Napisati program na programskom jeziku C koji pomaže enigmatičaru u sastavljanju ukrštenih reči. U tekst datoteci `ukrstenica.txt` se nalazi šema ukrštenice po sledećem formatu: u prvom redu se nalazi ceo broj N ($N \leq 20$) koji predstavlja dimenziju ukrštenice, a u narednih N redova se u svakom redu nalazi N celih brojeva koji označavaju da li je polje ukrštenice prazno (označava se nulom), gde se može pri popunjavanju ukrštenice upisati slovo, ili crni kvadrat (označava se jedinicom), gde se ne može upisati slovo. Brojevi su razdvojeni jednim blanko znakom. U datoteci `recnik.txt` se nalazi spisak reči koje enigmatičar želi da upotrebi u sastavljanju ukrštenice, jedna reč u svakom redu datoteke. Reči nisu duže od N karaktera. Program treba da na osnovu prve datoteke koja sadrži šemu ukrštenice utvrdi minimalnu i maksimalnu dozvoljenu dužinu reči koja se može smestiti u zadatu ukrštenicu po vertikali ili po horizontali, a zatim na standardnom izlazu ispiše sve reči iz zadatog spiska čija dužina se nalazi između minimalne i maksimalne dozvoljene dužine.

II PITANJA

1) Šta radi sledeća funkcija napisana na programskom jeziku C? Smatrati da su kapaciteti znakovnih nizova dovoljni za ispravan rad funkcije, kao i da su bajtovi posle kraja stringa popunjeni terminalnim karakterom ('0').

```
char *str(char *prvi, const char *drugi, int n) {
    int len = strlen(drugi); char *ret = prvi;
    while (*drugi) drugi++; drugi--;
    for (; *prvi; prvi++);
    n = n > len ? len : n;
    while (n--)
        if (!(*prvi++ = *drugi--)) return ret;
    return ret;
}
```

- A) Na kraj prvog stringa nadovezuje najviše prvih n karaktera drugog stringa u obrnutom poretku.
 B) Na kraj prvog stringa nadovezuje najviše poslednjih n karaktera drugog stringa u obrnutom poretku.
 C) Na kraj prvog stringa nadovezuje najviše poslednjih n karaktera drugog stringa

2) Šta ispisuje sledeći program na programskom jeziku C?

```
#include <stdio.h>
void zameni(int* a, int* b) {
    int t = *a; *a = *b; *b = t;
}
void preuredi(int arr[], int n, int i) {
    int m = i;
    int ls = 2*i + 1;
    int ds = 2*i + 2;
    if (ls < n && arr[ls] > arr[m]) m = ls;
    if (ds < n && arr[ds] > arr[m]) m = ds;
    if (m != i) {
        zameni(&arr[i], &arr[m]);
        preuredi(arr, n, m);
    }
}
void ispisi(int arr[], int n) {
    int i;
    for (i=0; i<n; ++i)
        printf("%d ", arr[i]);
}
int main() {
    int arr[] = {7, 12, 6, 5, 11, 13};
    int n = sizeof(arr)/sizeof(arr[0]);
    preuredi(arr, n, 0);
    ispisi(arr, n);
}
```

- A) 12 7 6 5 13 11 (B) 12 11 6 5 7 13 C) 13 11 6 5 7 12

3) Šta ispisuje sledeći program na programskom jeziku C? Funkcija `strchr` vraća pokazivač na prvo pojavljivanje zadatog znaka u zadatom stringu ili null ukoliko se znak ne pojavljuje u stringu. Program je pokrenut sledećom komandom: `./program aca milica jovan`

```
#include <stdio.h>
#include <string.h>
void f2(char *s) {
    do {
        putchar(s, stdout);
    } while (*s++ != ' ');
}
void f1(char *s){
    char *p = strchr(s, ' ');
    if (p == NULL) { printf("%s ", s);
    } else {
        f1(p + 1);
        f2(s);
    }
}
void main(int argc, char **argv) {
    char str[100], *pc;
    int i, j, n;
    j = 0;
    for(i=1; i<argc; i++){
        pc = argv[i];
        while(*pc){
            str[j++] = *pc++;
        }
        str[j++] = ' ';
    }
    str[j-1] = '\0';
    f1(str);
}
```

- (A) jovan milica aca (B) navoj acilim aca (C) aca acilim navoj

4) Neka se tip identifikatora `g` čita na sledeći način: `g` je funkcija koja vraća pokazivač na niz pokazivača na funkcije bez argumenta koje vraćaju pokazivač na `int`. Koja od navedenih deklaracija identifikatora `g` odgovara datom opisu?

- (A) `int *(*(*g())[])();` (B) `int (**g())[]();` (C) `int *(*(*g[]))();`