

Ponovljeni završni ispit iz Programiranja i programskog inženjerstva 4. srpnja 2006.

Napomene za sve zadatke:

- Nije dopušteno korištenje goto naredbe, te statičkih i globalnih varijabli.
- Studentima na pitanja ne smiju odgovarati asistenti koji ih čuvaju u učionici. Na pitanja će odgovarati isključivo asistenti koji za vrijeme ispita obilaze prostorije

1. (7 bodova) Napisati **program** koji generira 100 slučajnih cijelih brojeva iz zatvorenog intervala [1,36]. Nakon toga na zaslon treba ispisati sve brojeve koji su se pojavili više od 4 puta.

Primjer ispisa na zaslon:

```
Vise od 4 puta pojavili su se brojevi: 1 15 22 34 36
```

2. (8 bodova) Napisati **funkciju** kojoj se zadaju dva cjelobrojna jednodimenzionalna polja. Zadana polja ne moraju biti jednake duljine. Funkcija treba u pozivajući program vratiti vrijednosti najmanjeg elementa prvog polja i najvećeg elementa drugog polja (funkcija vraća dvije vrijednosti). Može se pretpostaviti da svako od zadanih polja sadrži barem jedan element (nije potrebno provjeravati). Također napisati glavni program (funkciju main) u kojem treba definirati i inicijalizirati dva cjelobrojna jednodimenzionalna polja (s duljinama i vrijednostima po želji), pozvati funkciju i na zaslon ispisati rezultat dobiven izvođenjem funkcije.

3. (8 bodova) Zapis postojeće direktne neformatirane (binarne) datoteke "catalog.bin" sadrži podatke o proizvodu: šifra proizvoda (int), naziv (niz znakova duljine 10+1), jedinična cijena (float). Redni broj zapisa u datoteci odgovara šifri proizvoda. Npr. u zapisu pod rednim brojem 512 nalaze se podaci za proizvod sa šifrom 512, nazivom "Vijak" i jediničnom cijenom 63.

Zapis postojeće slijedne formatirane (tekstualne) datoteke "naruci.txt" sadrži šifru proizvoda (cijeli broj) i broj naručenih komada proizvoda (cijeli broj).

Primjer sadržaja datoteke "naruci.txt":

```
512 2
64 1
714 3
```

Napisati **program** koji će stvoriti **novu** formatiranu (tekstualnu) datoteku "izlaz.txt" čiji zapis sadrži šifru i naziv proizvoda, broj naručenih komada, te jediničnu i ukupnu cijenu ispisane s točno dvije znamenke iza decimalne točke. Proizvodi u datoteci "izlaz.txt" moraju biti navedeni istim redoslijedom kojim su navedeni u datoteci "naruci.txt". U programu **nije potrebno** ispitivati uspješnost obavljanja operacija nad datotekama (npr. je li uspješno otvaranje, čitanje, pisanje, itd.).

Primjer sadržaja datoteke "izlaz.txt":

```
512 Vijak 2 63.00 126.00
64 Klin 1 121.50 121.50
714 Matica 3 1100.10 3300.30
```

4. (7 bodova) Napisati **funkciju** kojoj se zadaju nizovi znakova n_1 i n_2 . Funkcija treba sadržaj niza n_2 dodati (nadovezati) u niz n_1 , te u pozivajući program vratiti pokazivač na **posljednji** znak promijenjenog niza n_1 . Također, napisati glavni program koji će pozvati funkciju s vrijednostima iz dolje navedenog primjera, te pomoću dobivenog rezultata funkcije na zaslon ispisati posljednji znak promijenjenog niza.

Primjer: za zadane nizove "Uvijek" i "Nikad", funkcija prvi niz mijenja u "UvijekNikad" i vraća **pokazivač** na 11-ti znak prvog niza. Glavni program uz pomoć dobivenog pokazivača na zaslon ispisuje znak d.

1.

```
#include <stdlib.h>
#include <time.h>
#include <stdio.h>

int main () {
    int i, broj;
    int frek[36+1] = {0};
    srand ((unsigned) time(NULL));
    for (i = 0; i < 100; i++) {
        broj = (float) rand() / (RAND_MAX+1) * (36-1+1) + 1;
        frek[broj]++;
    }
    printf("Vise od 4 puta pojavili su se brojevi:");
    for (i = 1; i <= 36; i++)
        if (frek[i] > 4)
            printf(" %d", i);
    return 0;
}
```

2.

```
#include <stdio.h>

void vecimanji(int *p1, int dulj1, int *p2, int dulj2, int *najm, int *najv) {
    int i;
    *najm = p1[0];
    for (i = 1; i < dulj1; i++)
        if (p1[i] < *najm) *najm = p1[i];
    *najv = p2[0];
    for (i = 1; i < dulj2; i++)
        if (p2[i] > *najv) *najv = p2[i];
}

int main () {
    int polje1[] = {2, 1, 3};
    int polje2[] = {4, 5};
    int mali, veliki;
    vecimanji(polje1, 3, polje2, 2, &mali, &veliki);
    printf("Najmanji: %d    Najveci: %d\n", mali, veliki);
    return 0;
}
```

3.

```
#include <stdio.h>

int main () {
    FILE *katalogTok, *naruciTok, *izlazTok;
    int sifra, komada;
    struct {
        int sif;
        char naz[10+1];
        float jedCijena;
    } proiz;

    katalogTok = fopen("katalog.bin", "rb");
    naruciTok = fopen("naruci.txt", "r");
    izlazTok = fopen("izlaz.txt", "w");

    while (fscanf(naruciTok, "%d %d", &sifra, &komada) == 2) {
        fseek(katalogTok, (sifra-1)*sizeof(proiz), SEEK_SET);
        fread(&proiz, sizeof(proiz), 1, katalogTok);
        fprintf(izlazTok, "%d %s %d %.2f %.2f\n",
            proiz.sif, proiz.naz, komada, proiz.jedCijena, proiz.jedCijena*komada);
    }
    fclose(katalogTok);
    fclose(naruciTok);
    fclose(izlazTok);
    return 0;
}
```

4.

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

char *nadodaj (char *n1, char *n2) {
    strcat(n1, n2);
    return n1 + strlen(n1) - 1;
}

int main() {
    char n1[12] = "Uvijek";
    char *rez;
    rez = nadodaj(n1, "Nikad");
    printf("%c\n", *rez);
    return 0;
}
```