

2. međuispit iz predmeta Algoritmi i strukture podataka

7. svibnja 2008.

Napomena za sve zadatke:

- Nije dopušteno korištenje naredbe `goto`, te statičkih i globalnih varijabli.

Zadatak 1. (5 bodova)

Napišite **rekurzivnu** funkciju za sortiranje polja algoritmom sortiranja biranjem (*selection sort*). Prototip funkcije treba biti:

```
void RSort (int *polje, int n);
```

Za traženje indeksa najmanjeg člana u polju koristite funkciju *Najmanji*:

```
int Najmanji(int *polje, int n);
```

a za zamjenu vrijednosti dva elementa polja funkciju *Zamijeni*:

```
void Zamijeni(int *a, int *b);
```

Napomene:

Funkcije *Najmanji* i *Zamijeni* ne trebate implementirati. Nerekurzivno rješenje neće se priznavati.

Zadatak 2. (5 bodova)

Neka je zadan tip podatka *Stog* za koji su definirane funkcije za inicijalizaciju stoga, dodavanje elementa na stog te za brisanje elementa sa stoga. Prototipovi navedenih funkcija su:

```
void init_stog(Stog *stog);  
int dodaj(int element, Stog *stog);  
int skini(int *element, Stog *stog);
```

Funkcije *dodaj* i *skini* vraćaju 1 ako je operacija dodavanja odnosno skidanja uspjela, a 0 inače. Napišite funkciju čiji je prototip:

```
Stog Armstrong (Stog *stog, int *brojArmstrongovih);
```

koja mora vratiti novi stog koji se sastoji od svih troznamenskih Armstrongovih brojeva koji su se pojavili na ulaznom stogu *stog*. Broj je Armstrongov ako je jednak zbroju kubova svojih znamenki.

Ulazni stog sadrži samo troznamenkaste brojeve. Funkcija preko argumenta *brojArmstrongovih* mora vratiti ukupan broj pronađenih troznamenkastih Armstrongovih brojeva na stogu. Poredak brojeva u novom stogu nije bitan. Nakon završetka funkcije *Armstrong* sadržaj ulaznog stoga mora ostati nepromijenjen.

Zadatak 3. (5 bodova)

Zadan je niz brojeva: **10, 8, 6, 3, 5, 9, 1, 2, 4, 7**.

Ilustrirajte (ispišite niz nakon svake zamjene dvaju elemenata) **uzlazno** sortiranje zadanog niza brojeva

a) (2,5 boda) algoritmom **quicksort**. Stožer odaberite metodom aproksimacije medijana temeljem početnog, krajnjeg i srednjeg člana polja.

b) (2,5 boda) algoritmom **shellsort** za niz koraka {4, 3, 1}

Zadatak 4. (5 bodova)

Neka je zadan tip podatka Red za koji su definirane funkcije za inicijalizaciju reda, dodavanje elementa u red te za brisanje elementa iz reda. Prototipovi navedenih funkcija su:

```
void init_red(Red *red);  
int dodaj(int element, Red *red);  
int skini(int *element, Red *red);
```

Funkcije dodaj i skini vraćaju 1 ako je operacija dodavanja odnosno skidanja uspjela, a 0 inače. Napišite funkciju čiji je prototip:

```
void IzbaciUzastopne(Red *red);
```

koja će u zadanom redu ukloniti sva uzastopna ponavljanja brojeva.

Primjer: Red brojeva 3 3 5 6 7 7 12 5 5 5 5 7 7 7 7 5 funkcija će promijeniti u 3 5 6 7 12 5 7 5.

Rješenja

1. zadatak

```
void RSort (int *polje, int n)
{
    int indeks;
    if (n > 0)
    {
        indeks = Najmanji(polje, n);
        Zamijeni(&polje[0], &polje[indeks]);
        RSort (polje+1, n-1);
    }
}
```

2. zadatak

```
Stog Armstrong (Stog *stog, int *brojArmstrongovih) {

    Stog novi, temp;
    int broj;

    init_stog(&novi);
    init_stog(&temp);
    *brojArmstrongovih = 0;

    while (skini(&broj, stog)) {
        dodaj(broj, &temp);
        if (broj == (pow(broj/100,3) + pow((broj/10)%10,3) + pow(broj%10,3))) {
            dodaj(broj, &novi);
            (*brojArmstrongovih)++;
        }
    }
    while (skini (&broj, &temp)) dodaj (broj, stog);

    return novi;
}
```

3. zadatak

a)

<u>10</u>	8	6	3	<u>5</u>	9	1	2	4	7
5	8	<u>6</u>	3	10	9	<u>1</u>	2	4	7
5	8	1	<u>3</u>	10	9	6	<u>2</u>	4	7
<u>5</u>	8	1	2	<u>10</u>	9	6	3	<u>4</u>	7
4	<u>8</u>	1	2	5	<u>9</u>	6	3	10	<u>7</u>
<u>4</u>	7	1	<u>2</u>	5	8	6	3	10	9
2	<u>7</u>	1	4	<u>5</u>	8	6	3	10	9
2	<u>5</u>	1	4	<u>7</u>	8	6	<u>3</u>	10	9
2	<u>3</u>	<u>1</u>	4	5	8	6	7	10	9
<u>2</u>	<u>3</u>	3	4	5	8	6	7	10	9
1	2	3	4	5	<u>8</u>	<u>6</u>	7	10	9
1	2	3	4	5	6	<u>8</u>	<u>7</u>	10	9
1	2	3	4	5	6	7	8	<u>10</u>	<u>9</u>
1	2	3	4	5	6	7	8	9	10

b)

10 8 6 3 **5** 9 1 2 4 **7**

stožer je: 7

5 **8** 6 3 4 9 1 **2** 7 10

5 2 6 3 4 **9** **1** 8 7 10

5 2 6 3 4 1 **9** 8 **7** 10

5 2 6 3 4 1 7 8 9 10

5 2 **6** 3 4 **1**

stožer:5

1 2 4 3 5 6

1 **2** 4 3

stožer:2

1 4 2 3

1 2 **4** **3**

1 2 3 4 5 6 7 8 9 10

4. zadatak

```
void IzbaciUzastopne(Red *red){
    Red pom; int element, prethodni;
    init_red(&pom);

    if (skini(&element, red)){
        prethodni = element;
        dodaj(element, &pom);
        while(skini(&element, red)){
            if (element != prethodni)
                dodaj(element, &pom);
            prethodni = element;
        }
        while(skini(&element, &pom)){
            dodaj(element, red);
        }
    }
}
```