

Ex1 *

Daca alegeti optiunea sa incarcati fisier scrieti cratima ("-") la raspuns, aici in formular.

1. Scrieti cate un program care face suma elementelor unui vector folosind structura repetitiva indicata:

- a. for ... do **0.5p**
- b. while ... do **0.5p**
- c. O **functie** recursiva **1p**

-

Ex 2 *

Daca alegeti optiunea sa incarcati fisier scrieti cratima ("-") la raspuns, aici in formular.

2. Rezolvati urmatoarele recurente folosind Teorema Master:

- a. $T(n) = T(n/2) + n$ **1p**
- b. $T(n) = 16 * T(n) + n^2$ **0.5p**
- c. $T(n) = 0.25 * T(n/4) + \log(n)$ **0.5p**

-

Ex 3 *

Daca alegeti optiunea sa incarcati fisier scrieti cratima (" ") la raspuns, aici in formular.

3.3.5p

a. Demonstrati complexitatea in urmatoarea secventa de cod (ne uitam la instructiunea care se executa de cele mai multe ori): **2.25p**

```
1. sum <- 0
2. i <- 1
3. while i < n * n do
4.     for j <- i + 1 to n * n do
5.         for k <- j down to 0 do
6.             sum <- sum + 1
7.     i <- i + 1
```

b. Daca modificam for-ul de pe linia 4 sa fie
4. for j <- i + 1 to n do

Cum se modifica suma? Linia 5 se executa la fel de multe ori? Daca raspunsul este nu, motivati de ce. Hint: incercati sa separati suma in 2, o parte unde este indeplinita conditia din for-ul de pe linia 4 si o parte unde nu mai este indeplinita. Nu este nevoie sa calculati efectiv suma ci o puteti scrie si explica pe baza ei cum se modifica pasii. **1.25p**

4. 2.5p

a. Rezolvati operatia de recurenta folosind metoda iteratiei, porning de la formula matematica cu cazul de baza si cazul general apoi demonstrati complexitatea obtinuta. **2p**

Fie x,n doua numere intregi:

```
function the_sum(x, n)
begin
    if(x==0) then
        return 0;
    if(n %2 == 0) then
        return 1 + the_sum(x/3, n);
    return 1 + the_sum(x/3, n);
end
```

b. Ce face de fapt functia? **0.5p**