

Programmazione dei Calcolatori con Laboratorio

Esame del 23 settembre 2013 per informatici

Si vuole implementare una funzione che crei una lista contenente gli interi di una progressione aritmetica generata da tre interi (i, j, k) : l'intero i identifica il primo elemento della progressione; l'intero j identifica il termine della progressione e l'intero k specifica la differenza tra due elementi consecutivi della progressione.

Se $k > 0$, la progressione aritmetica generata da (i, j, k) è $i, i + k, i + 2k, \dots, i + tk$ dove t è il massimo intero per cui $i + tk < j$. Altrimenti, se $k < 0$ la progressione aritmetica generata da (i, j, k) è $i, i + k, i + 2k, \dots, i + tk$ dove t è il massimo intero per cui $i + tk > j$. Se $k = 0$ la progressione aritmetica è vuota.

Ad esempio se $i = 1, j = 10, k = 2$ la progressione aritmetica è $[1, 3, 5, 7, 9]$; se $i = 1, j = 1, k = 1$ la progressione aritmetica è vuota; se $i = 2, j = -3, k = -1$ la progressione aritmetica è $[2, 1, 0, -1, -2]$.

Si implementi in C una funzione denominata `Range()`, che prende in input una lista `a` e tre interi i, j e k e restituisce la lista `a` contenente la progressione aritmetica generata dalla tripla (i, j, k) .

Si calcoli il costo della funzione sia in termini di tempo che di memoria supplementare utilizzata.

Nel risolvere il problema si tenga conto che una lista è un puntatore a `nodo` (il primo nodo della lista) e che il tipo `nodo` è definito come segue.

```
struct nodo {
    int inf;
    struct nodo *succ;
    struct nodo *prec;
};
typedef struct nodo nodo;
```

Si ricorda che il campo `inf` contiene il dato del nodo e i campi `succ` e `prec` fanno riferimento rispettivamente al nodo successivo e a quello precedente.

Modalità di consegna: Lo studente deve consegnare **un unico** file denominato `CognomeNome.c` (dove `Cognome` e `Nome` stanno rispettivamente per il proprio cognome ed il proprio nome). Tale file deve contenere soltanto:

- la funzione richiesta (`Range()`) che a sua volta deve rispettare le specifiche imposte dal problema;
- la definizione delle strutture dati e tipi eventualmente utilizzati (compreso il `nodo`) e le funzioni che le utilizzano nonché gli *header* delle librerie utilizzate.
- ogni altra funzione utilizzata dalla soluzione.

La funzione `main()` **non** deve essere inclusa nel file `CognomeNome.c` pertanto si consiglia di definirla in un secondo file denominato `main.c`. I due file possono essere compilati insieme utilizzando il comando

```
gcc main.c CognomeNome.c
```