

Programmazione dei Calcolatori con Laboratorio

Esame del 9 luglio 2013 per informatici

La *porzione* di una lista **a** è un segmento della lista composta da elementi contigui di **a**. Si vuole implementare una funzione denominata `Porzione()` che prende in input la lista **a** e due interi **i** e **j**. Questa deve restituire una copia della porzione di **a** racchiusa tra l'elemento in posizione **i** e quello in posizione **j** escluso (si adotta la convenzione che il primo elemento della lista è in posizione 0). Inoltre, se $j = 0$ o se **j** supera la lunghezza della lista si deve restituire la porzione di lista che va da **i** in poi. Infine se **i** supera la lunghezza della lista oppure se $i \geq j$ si deve restituire la lista vuota.

Ad esempio se **a** = [12, 14, 14, 15, 3, 3, 6, 8], `Porzione(a, 1, 4)` deve restituire la nuova lista [14, 14, 15] mentre `Porzione(a, 0, 1)` deve restituire la lista [12]. Infine `Porzione(a, 5, 0)` deve restituire [3, 6, 8].

Si implementi in C la funzione `Porzione()`, che prende in input la lista **a** e i due interi **i** e **j** e restituisce la copia della porzione di **a** individuata dai due interi.

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 (`Porzione()`) 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
```