

Esercitazione 8

Argomento: SCL

Per questa esercitazione l'organizzazione del codice nei vari file e il testing delle funzioni è lasciata agli studenti.

Scaricate la cartella es8 e completate i file forniti. In particolare:

- l'implementazione delle funzioni va scritta nel file `functions.c`
- le intestazioni delle funzioni va messa nel file `functions.h`
- il main è nel file `test.c`
- I file `scl.c` e `scl.h` contengono implementazione e intestazioni di funzioni utili per lavorare con le scl, che potete usare senza doverle re-implementare (es creazione scl random, stampa scl etc).

Compilate con l'istruzione

make

Eseguite l'eseguibile 'test'

./test

Esercizio 8.1

Sia data la seguente struttura collegata `TipoSCL`:

```
typedef float TipoInfoSCL;
struct ElemSCL {
    TipoInfoSCL info ;
    struct ElemSCL* next;
};
typedef struct ElemSCL TipoNodoSCL;
typedef TipoNodoSCL * TipoSCL;
```

Implementare la funzione:

```
TipoSCL doubledCopy( TipoSCL list );
```

che data una SCL di input, allochi e restituisca il puntatore ad una nuova SCL che contiene, per ogni elemento della SCL di input, due ripetizioni contigue dello stesso. L'ordine degli elementi ripetuti dovrà essere quello della SCL di input.

Si gestisca il caso in cui la SCL di input è nulla, restituendo un puntatore nullo. La SCL in ingresso alla funzione non deve essere modificata in alcun modo.

Esercizio 8.2

Implementare la funzione:

```
TipoSCL multipleSubset( TipoSCL list, unsigned int m );
```

che data in input una SCL list, restituisca il puntatore ad una nuova SCL che contiene i soli elementi di list i cui valori sono multipli di m (si usi la convenzione in cui lo 0 è multiplo di tutti i numeri interi). L'ordine degli elementi della SCL restituita dovrà essere quello della SCL di input.

Si gestisca il caso in cui il puntatore list e' nullo, restituendo un puntatore nullo. La lista in ingresso alla funzione non deve essere modificata in alcun modo.

Esercizio 8.3

Implementare la seguente funzione C:

```
TipoSCL pairSubset( TipoSCL list );
```

che data in input una SCL list, restituisca il puntatore ad una nuova SCL che contiene i soli elementi di list il cui indice è pari, assumendo che il primo elemento abbia indice 0. L'ordine degli elementi della SCL di output dovrà essere quello della SCL di input.

Si gestisca il caso in cui il puntatore list e' nullo, restituendo un puntatore nullo. La SCL in ingresso alla funzione non deve essere modificata in alcun modo.

Esercizio 8.4

Implementare la seguente funzione C:

```
TipoSCL getPrev(TipoSCL list, TipoSCL n);
```

che data in input una SCL list e il puntatore n a un elemento della lista, restituisca il puntatore all'elemento precedente di n.

Esercizio 8.5

Implementare la seguente funzione C:

```
TipoSCL getTail(TipoSCL list);
```

che data in input una SCL list, restituisca il puntatore all'ultimo elemento della lista.