Fondamenti di Informatica
Esempio programmazione 3

Istruzioni per lo svolgimento e la consegna

- La prima operazione da effettuare è modificare il file `studente.txt` (presente nella directory dove avete trovato questo `pdf`) inserendo il proprio nome e cognome e numero di matricola. Utilizzare un semplice editor di testo, salvando il file senza modificarne il nome.
- Nella stessa directory sono presenti i file necessari allo svolgimento dell'esercizio. **Per ogni esercizio dovrà essere modificato solamente il file `.c` corrispondente. Non devono essere modificati né spostati o eliminati i rimanenti file, pena la valutazione negativa dell'assegnamento**. Nel file `.c` dovranno essere mantenuti tutti e soli gli output a schermo forniti, in modo da mantenere la corrispondenza con l'output di esempio. Inoltre, è possibile definire funzioni ausiliarie all'interno del sorgente `.c`.
- Per compilare e generare l'eseguibile, da terminale entrate nella directory dove avete trovato questo `pdf` e lanciate il comando `make nome_esercizio`. Il `nome_esercizio` corrisponde al nome del sorgente, privato dell'estensione `.c`. Verrà generato l'eseguibile `nome_esercizio` che, lanciato da terminale (`./nome_esercizio`), vi permetterà di provare il vostro programma.
- Lanciando invece il comando (`./self_evaluation nome_esercizio`) eseguirete in maniera automatica alcuni test per verificare le soluzioni che avete implementato. I test sono studiati per verificare anche i casi particolari, in modo da gestire quelli che possono essere errori comuni in fase di implementazione. **Tenete presente che il correttore funziona solo all'interno di una distribuzione Linux a 64 bit (ad esempio, le macchine messe a disposizione nel laboratorio)**.
- La procedura di consegna dovrà iniziare lanciando il programma `./prepara_consegna.sh` presente nella directory dove avete trovato il presente `pdf`. Una volta lanciato, esso genererà un archivio di nome `consegna.tar.gz`: tale file sarà **l'unico che dovrà essere inviato** attraverso il sito `https://stem.elearning.unipd.it` per consegnare il vostro elaborato, seguendo anche le istruzioni che saranno fornite in aula dai docenti.

Considerate 2 aspetti:

- Se ci sono errori in compilazione/esecuzione, c'è qualcosa che rende errata/incompleta la vostra implementazione;
- Se non ci sono errori in compilazione/esecuzione, verificate che i risultati siano corretti (in alcuni casi è molto semplice fare il calcolo anche a mente).

1 Liste medie (`list_average.c`)

Data una lista concatenata di `int`, sviluppare e testare le seguenti funzioni C:

```c
// Calcola la media dei valori contenuti nella lista in input
float get_list_average(struct Node* head);

// Crea una nuova lista composta dai nodi della lista in input
// il cui valore supera la media della dati contenuti nella lista
struct Node* list_average(struct Node *head);
```

Completare le funzioni sapendo che:

- `head` è il puntatore alla lista che contiene i dati da esaminare all'interno della funzione;
- la nuova lista deve essere allocata dinamicamente;
- la nuova lista avrà al massimo tanti nodi quanti la lista iniziale (normalmente meno);
- i nodi della nuova lista saranno creati a partire dai dati presenti nella lista puntata da `head` solo nel caso in cui il valore del nodo sia maggiore della media dei valori contenuti nella lista `head`;
- se la lista in input è vuota ritornare `NULL`.

Il programma di test è già fornito in forma di file sorgente `list_average_main.c`. Per riuscire ad utilizzare usare il comando `gcc`, è necessario compilare i file sorgenti come segue:

```
gcc -o list_average list_average.c list_average_main.c
```

Esempio 1

Lista in input:

```
21 -> -55 -> 69 -> -14 -> NULL
```

Lista in output:

```
21 -> 69 -> NULL
```

Esempio 2

Lista in input:

```
2 -> 2 -> 0 -> NULL
```

Lista in output:

```
2 -> 2 -> NULL
```

2 Da lista ad array (`list2array.c`)

Data una lista concatenata di `int`, sviluppare e testare la seguente funzione **ricorsiva** C:

```
// Copia la lista puntata da head nell'array preallocato output
// La funzione restituisce il numero di elementi inseriti nell'array
// La funzione deve essere implementata in maniera ricorsiva
int list2array(struct Node* head, int *output);
```

Completare le funzioni sapendo che:

- `head` è il puntatore alla lista che contiene i dati da esaminare all'interno della funzione;
- il vettore `output` è già preallocato e ha spazio sufficiente a contenere tutti gli elementi della lista;
- il numero effettivo di elementi copiati nell'array sarà lo stesso dei nodi della lista in input;
- se la lista in input è vuota, lasciare inalterato `output` e ritornare 0.

Il programma di test è già fornito in forma di file sorgente `list2array_main.c`. Per riuscire ad utilizzare usare il comando gcc, è necessario compilare i file sorgenti come segue:

```
gcc -o list2array list2array.c list2array_main.c
```

Esempio 1

Lista in input:

```
21 -> -55 -> 69 -> -14 -> NULL
```

Array in output:

```
[ 21 -55 69 -14 ]
```

Esempio 2

Lista in input:

```
2 -> 2 -> 0 -> NULL
```

Array in output:

```
[ 2 2 0 ]
```