

# Fondamenti di Informatica

## Esempio programmazione 1

### 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 normalizzate (**list\_normalize.c**)

Data una lista concatenata di double, sviluppare e testare le seguenti 3 funzioni C:

---

```
// Trova il minimo valore nella lista concatenata
bool min_data_list(struct Node *head, double* min);
// Trova il massimo valore nella lista concatenata
bool max_data_list(struct Node *head, double* max);
// Normalizza gli elementi della lista
void normalize_list(struct Node *head, double min, double max);
```

---

Completare le funzioni in modo che in fase di normalizzazione:

- il massimo (*max*) degli elementi della lista assuma il valore 1;
- il minimo (*min*) degli elementi della lista assuma il valore 0;
- i valori rimanenti (*x*) mantengano la distribuzione tra *min* e *max* ( $0 \leq x \leq 1$ ) con la formula

$$x = \frac{x - \min}{\max - \min}.$$

Il programma di test è già implementato e compilato come file oggetto `list_normalize_main.obj`. Per riuscire ad utilizzare usare il comando **gcc**, è necessario linkare il file come segue:

---

```
gcc -o list_normalize list_normalize.c list_normalize_main.obj
```

---

### Esempio 1

Lista in input:

---

2.1 -> -5.5 -> 6.9 -> -1.4 -> NULL

---

Lista in output:

---

0.612903 -> 0 -> 1 -> 0.330645 -> NULL

---

### Esempio 2

Lista in input:

---

2 -> 2 -> 0 -> NULL

---

Lista in output:

---

1 -> 1 -> 0 -> NULL

---

## 2 Capitalize (**capitalize.c**)

Scrivere una funzione C che data una stringa renda maiuscole tutte le lettere iniziali delle parole più lunghe di un carattere e minuscole le altre lettere.

Le parole sono composte esclusivamente da lettere, spazi ( ' ' ), e terminatori ('\0').

Le modifiche devono avvenire direttamente nella stringa in ingresso.

L'operazione si conclude una volta raggiunta la fine della stringa ('\0').

La funzione da sviluppare è la seguente:

---

```
// Rende maiuscole tutte le lettere iniziali delle parole
// di lunghezza maggiore di un carattere e minuscole le altre lettere
void capitalize(char *str);
```

---

Il programma di test è già implementato e compilato come file oggetto `capitalize_main.obj`. Per riuscire ad utilizzare usare il comando `gcc`, è necessario linkare il file come segue:

---

```
gcc -o capitalize capitalize.c capitalize_main.obj
```

---

### Esempio 1

Stringa in input:

---

```
Ho attraversato gli oceani del tempo per trovarti
```

---

Stringa in output:

---

```
Ho Attraversato Gli Oceani Del Tempo Per Trovarti
```

---

### Esempio 2

Stringa in input:

---

```
Fatti non foste a viver come BRUTI ma per seguir virtute e canoscenza
```

---

Stringa in output:

---

```
Fatti Non Foste a Viver Come Brutì Ma Per Seguir Virtute e Canoscenza
```

---