La presente esercitazione presuppone conoscenza delle seguenti parti del linguaggio C: *main*, *return*, *commenti*, *variabili*, *identificatori*, *tipi scalari*, *assegnazione*, *costanti letterali*, *printf (base)*, *scanf (deprecata)*, *operazioni aritmetiche*, *if ... else*, *operatori relazionali*, *operatori Booleani*, *getchar*, *putchar*, *ciclo while*, *libreria ctype.h*, *const*, *#define*, *exit*, *programmazione a stati*, *gets*, *sscanf*, *stringhe*, *traduzione di algoritmi matematici*; *switch*, *break*, *#ifdef*, *math.h*, *cast*, *printf (completa)*, *ciclo for*; *argc*, *argv*; *definizione di funzioni*; *file sequenziali*, *fopen*, *fclose*, *fgetc*, *fprintf*

Esercizio 1

Scrivere un programma in linguaggio C che legga valori di temperatura (uno per riga finché ce ne sono) in gradi Fahrenheit e ne restituisca il valore in gradi Celsius. Per la risoluzione dell'esercizio è obbligatorio scrivere ed utilizzare una funzione `celsius` così definita:

```
float celsius (float temperatura_F);  
/* riceve come parametro la temperatura in gradi Fahrenheit */  
/* restituisce la stessa temperatura espressa in gradi Celsius */
```

Si ricorda che le temperature in gradi Fahrenheit (t_F) e Celsius (t_C) sono legate dalla seguente relazione:

$$t_C = (t_F - 32) \cdot 5/9$$

Esercizio 2

Scrivere un programma in linguaggio C che legga valori di temperatura (uno per riga finché ce ne sono) in gradi Celsius e ne restituisca il valore in gradi Fahrenheit. Per la risoluzione dell'esercizio è obbligatorio scrivere ed utilizzare una funzione `fahrenheit` così definita:

```
float fahrenheit (float temperatura_C);  
/* riceve come parametro la temperatura in gradi Celsius */  
/* restituisce la stessa temperatura espressa in gradi Fahrenheit */
```

Esercizio 3

Scrivere un programma in linguaggio C che riceva da riga di comando un numero intero positivo inferiore a 1000 e determini se tale numero è primo oppure no. Per la risoluzione dell'esercizio è obbligatorio scrivere ed utilizzare una funzione `is_prime` così definita:

```
int is_prime (unsigned long n);  
/* riceve come parametro il numero da valutare */  
/* restituisce 0 se non e' un numero primo, 1 altrimenti */
```

Si ricorda che un numero è primo se è maggiore di 1 e divisibile esattamente solo per l'unità e per se stesso. Ne consegue che per sapere se un numero maggiore di 1 non è primo è sufficiente controllare se è divisibile esattamente per due o per uno dei numeri dispari non superiori alla radice quadrata del suo valore.

Esercizio 4

Definire in linguaggio C una funzione denominata `esame_superato` che riceva come parametri due voti (sotto forma di numeri interi `vteoria` e `vprog`) e fornisca come valore di ritorno:

- -1 se è stato fornito un parametro errato (ossia un voto negativo o superiore a 30)
- 0 se l'esame non è stato superato
- il valore del voto conseguito se l'esame è stato superato

In altre parole la funzione deve avere il seguente prototipo:

```
int esame_superato (int vteoria, int vprog)
```

Vale la seguente regola per il superamento dell'esame:

L'esame è superato se la media aritmetica dei due voti (arrotondata per eccesso) è pari almeno a 18 e nessuno dei due voti è inferiore a 15.

Scrivere quindi un programma in linguaggio C che (usando la funzione `esame_superato`) indichi il superamento dell'esame in base ai due voti ricevuti sulla riga di comando.

Il seguente esempio riporta tre casi di uso del programma (supposto essere denominato `voto`):

<i>attivazione</i>	<i>output</i>
voto 24 26	Complimenti, hai superato l'esame col voto 25/30!
voto 12 30	Mi spiace, esame non superato.
voto -1 30	errore - dati non validi.

Esercizio 5

Scrivere un programma in linguaggio C che calcoli quante righe sono presenti nel file denominato `DATI.TXT`. Si osservi che la massima lunghezza di una riga del file non è nota a priori.

Esercizio 6

Scrivere un programma in linguaggio C che calcoli quanti caratteri e quante righe sono presenti nel file il cui nome è indicato come primo parametro sulla riga di comando.

Ad esempio se il programma (denominato `carrig`) venisse attivato col comando:

```
carrig lettera.txt
```

ed il file `lettera.txt` contenesse il seguente testo:

```
Cara mamma,  
torno per Natale.  
Baci.
```

allora il programma dovrebbe fornire in output:

```
lettera.txt: caratteri=33 righe=3
```

Esercizio 7

Un file contiene i prezzi degli articoli presenti a magazzino, indicati tramite il codice dell'articolo (numero intero positivo) ed il suo prezzo in Euro (numero reale con due cifre frazionarie). Ogni riga del file contiene un articolo.

Scrivere un programma in linguaggio C che applichi ai prezzi (contenuti nel file il cui nome è passato come primo parametro sulla riga di comando) uno sconto pari alla percentuale indicata come secondo parametro sulla riga di comando, scrivendo il risultato in un file il cui nome è passato come terzo parametro sulla riga di comando.

Ad esempio se il programma (denominato `varprez`) venisse attivato col comando:

```
varprez pnormali.txt 20 pscontati.txt
```

ed il file `pnormali.txt` contenesse i seguenti dati:

```
1 1.00  
10100 20.00  
1234 5.50
```

allora il programma dovrebbe generare il file `pscontati.txt` coi seguenti dati:

```
1 0.80  
10100 16.00  
1234 4.40
```