

Nota

È considerato errore qualsiasi output non richiesto dagli esercizi.

Esercizio 1 (punti 6)

Creare i file `vettori.h` e `vettori.c` che consentano di utilizzare la seguente funzione:

```
extern double prodotto_scalare (const double *x, const double *y, size_t n);
```

La funzione accetta come parametri due puntatore a vettori di `double` `x` e `y` e un dato di tipo `size_t` che ne indica la dimensione `n`.

La funzione restituisce il prodotto scalare dei due vettori:

$$\langle x, y \rangle = \sum_{i=0}^{n-1} x_i y_i$$

Se `n` è 0 o se uno dei due puntatori è `NULL`, la funzione deve ritornare 0.

Esercizio 2 (punti 7)

Nel file `trigonometria.c` implementare in linguaggio C la funzione corrispondente alla seguente dichiarazione:

```
extern double coseno(double x);
```

La funzione deve calcolare il valore di $\cos(x)$ utilizzando il seguente sviluppo in serie di Taylor:

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n}$$

Non è consentito l'uso di librerie esterne.

Esercizio 3 (punti 7)

Nel file `stringhe.c` implementare la definizione della funzione:

```
extern char *parola_piu_lunga(const char *sz);
```

La funzione accetta come parametro una stringa zero terminata e deve restituire un puntatore ad una nuova stringa zero terminata (allocata dinamicamente nell'heap) contenente una copia della parola più lunga presente all'interno della stringa `sz`, dove con "parola" intendiamo una sequenza contigua di caratteri diversi da spazio. Se ci sono più parole della lunghezza massima, la funzione ritorna la prima incontrata.

Esercizio 4 (punti 7)

Creare i file `righe.h` e `righe.c` che consentano di utilizzare la seguente funzione:

```
extern unsigned int conta_righe(FILE *f);
```

La funzione accetta come parametro un puntatore a un file aperto in modalità `tr` (testo) e deve ritornare il numero di righe del file. Un file vuoto o senza caratteri <a capo> ha una riga. Non è possibile imporre alcun vincolo sulla massima lunghezza delle righe.

Esercizio 5 (punti 6)

Nel file `cornici.c` implementare la definizione della funzione:

```
extern void cornici(unsigned int n);
```

La funzione deve inviare a `stdout` un `+` circondato da `n` cornici con `'|'` sui lati verticali e `'-'` sui lati orizzontali. Ad esempio chiamando la funzione con `n=0`, la funzione deve inviare su `stdout` solo il centro:

`+`

Chiamando la funzione con `n=2`, la funzione deve inviare su `stdout`:

```
+---+
|+-+|
||+||
|+-+|
+---+
```

Ovvero (visualizzando ogni carattere in una cella della seguente tabella):

+	-	-	-	+	<a capo>
	+	-	+		<a capo>
		+			<a capo>
	+	-	+		<a capo>
+	-	-	-	+	<a capo>