

Esercizio 1

Creare un file “main.c”. Nel file, si completi la seguente funzione utilizzando comandi del linguaggio C:

```
double altezza_caduta (double g, double t) {  
    ...  
}
```

La funzione calcola l'altezza di un terrazzo, dato il tempo impiegato da un sasso a raggiungere il suolo dopo essere stato lasciato cadere. Si utilizzi la formula:

$$h = \frac{1}{2}gt^2$$

Esercizio 2

Creare un file “main.c”. Nel file, si completi la seguente funzione utilizzando comandi del linguaggio C:

```
double serie(unsigned int n) {  
    ...  
}
```

La funzione calcola il risultato della sommatoria:

$$\sum_{k=1}^n \frac{1}{k^2}$$

Esercizio 3

Spostare la funzione dell'esercizio 2 in un altro file C chiamato “serie.c”, creare il corrispondente header “serie.h” (che conterrà solo le dichiarazioni) e includerlo nel file “main.c”. Ricordate di inserire nel file .h l'include guard e nel file .c di includere l'header corrispondente. Verificare il funzionamento e il comportamento con il debugger.

Esercizio 4

Creare un file “main.c”. Nel file, si completi la seguente funzione utilizzando comandi del linguaggio C:

```
uint8_t *tabellina(uint8_t n) {  
    ...  
}
```

La funzione alloca dinamicamente nella memoria heap un vettore di 10 interi senza segno a 8 bit, e lo riempie con i valori della tabellina di n (da $n \times 1$ a $n \times 10$). Infine, la funzione restituisce il puntatore all'area di memoria allocata. Se $n > 12$, la funzione restituisce NULL.

Esercizio 5

Creare un file “main.c”. Nel file, si completi la seguente funzione utilizzando comandi del linguaggio C:

```
char *sostituisci_spazi(const char *s, char c) {  
    ...  
}
```

I parametri della funzione sono un puntatore al primo elemento di un vettore di `char` zero terminato `s` (stringa C) e un carattere `c`. La funzione deve restituire il puntatore al primo elemento di un nuovo vettore di `char` allocato dinamicamente sulla memoria heap, contenente una copia della stringa `s` in cui tutte le occorrenze del carattere `<spazio>` sono sostituite dal carattere `c`. Se `s` vale NULL, la funzione restituisce NULL.

Esercizio 6

Creare un file “main.c”. Nel file, si completi la seguente funzione utilizzando comandi del linguaggio C:

```
uint32_t conta_doppie(const char *s) {  
    ...  
}
```

La funzione accetta come parametro un puntatore al primo elemento di un vettore di `char` zero terminato (stringa C), e restituisce il numero di occorrenze di lettere doppie al suo interno. Sono considerate lettere tutti i caratteri ASCII appartenenti all’alfabeto latino, e `s` non conterrà mai lettere triple. Se `s` vale `NULL`, la funzione restituisce 0. Ad esempio, se `s` = “La mamma ha fatto i toast”, la funzione deve restituire 2.