

Esercizio 1

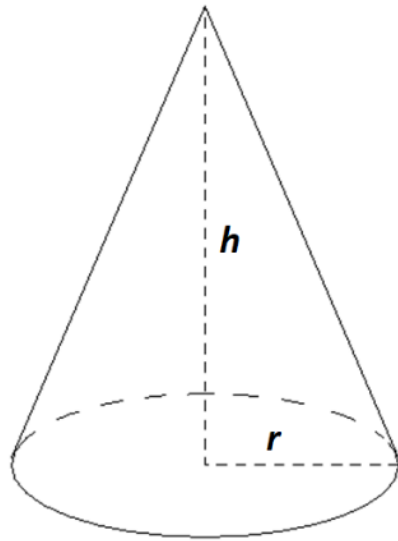
Creare un file “main.c”. Nel file, si completi la seguente funzione utilizzando comandi del linguaggio C:

```
double volume_cono (double r, double h) {  
    ...  
}
```

La funzione deve, in caso che r o h siano minori di 0, ritornare il valore 0.

Dopo aver effettuato i dovuti controlli, la funzione deve calcolare il volume di un cono dati l'altezza e il raggio, secondo la formula:

$$V = \frac{\pi r^2 h}{3}$$



Esercizio 2

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

```
uint32_t somma_n (uint32_t n) {  
    ...  
}
```

La funzione deve restituire la somma dei numeri naturali fino a n incluso.

Esercizio 3

Spostare la funzione dell'esercizio 2 in un altro file C chiamato “somma_n.c”, creare il corrispondente header “somma_n.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 col debugger.

Esercizio 4

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

```
void somma_n (uint32_t *n) {  
    ...  
}
```

Che sostituisca al valore nella cella di memoria a cui punta n , il valore calcolato secondo la funzione descritta nell'esercizio precedente.

Esercizio 5

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

```
int *are_multiple (int *v, size_t size, int n) {  
    ...  
}
```

La funzione accetta come argomenti un vettore di interi con segno a 32 bit v di dimensione $size$. La funzione deve allocare dinamicamente nella memoria heap un nuovo vettore di interi con segno a 32 bit di dimensione $size$, che contenga 1 nelle posizioni in cui v contiene un multiplo di n , 0 nelle altre. La funzione ritorna un puntatore al vettore così creato.

Esercizio 6

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

```
int *merge (int *src1, int *src2, size_t size) {  
    ...  
}
```

La funzione accetta come argomenti due vettore di interi con segno a 32 bit, $src1$ e $src2$, entrambi di dimensione $size$, e deve ritornare un puntatore ad un vettore di interi con segno a 32 bit di dimensione $size*2$ allocato dinamicamente nella memoria heap. Il vettore così creato deve contenere i valori di $src1$ e $src2$ con ordine alternato.

Esempio:

src1:	1	3	5	7				
src2:	2	4	6	8				
res:	1	2	3	4	5	6	7	8

Esercizio 7

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

```
uint32_t fibonacci (uint32_t n) {  
    ...  
}
```

La funzione deve restituire il termine n-esimo della successione di Fibonacci, che è definita come

$$a_0 = 0$$

$$a_1 = 1$$

$$a_n = a_{n-1} + a_{n-2}, n \in \mathbb{N} \wedge n \geq 2$$