

LEGGETE LA GUIDA PER LA CREAZIONE DEI PROGETTI E PER IL DEBUGGING!

*Gli esercizi seguenti devono essere risolti, compilati e testati utilizzando il debugger. Per ognuno si deve realizzare una funzione `main()` che ne testi il funzionamento. **Fate progetti diversi per ogni esercizio.***

Esercizio 1

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

```
int primo (unsigned int val) {  
    ...  
}
```

La funzione riceve un numero intero `val` e restituisce 1 se `val` è primo, 0 altrimenti. Se non conoscete la definizione di numero primo, usate Wikipedia.

Esercizio 2

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

```
unsigned int prossimo_numero_primo(unsigned int x) {  
    ...  
}
```

La funzione deve ritornare il più piccolo numero primo maggiore di `x`.

Esercizio 3

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

```
unsigned int somma_cifre(unsigned int x) {  
    ...  
}
```

La funzione deve restituire la somma delle cifre della rappresentazione in base 10 di `x`. Ad esempio, se `x` vale 123, la funzione deve restituire 6.

Esercizio 4

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

```
int divisione(double a, double b, double *q) {  
    ...  
}
```

La funzione accetta due `double` `a` e `b`. Se `b` è diverso da 0, mette nella variabile puntata da `q` il risultato di `a` diviso `b` e ritorna 1, altrimenti ritorna 0 e non modifica la variabile puntata da `q`.

Esercizio 5

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

```
double discriminante (double a, double b, double c) {  
    ...  
}
```

La funzione deve calcolare il discriminante dell'equazione di secondo grado la cui formula è:

$$ax^2 + bx + c = 0.$$

Il discriminante è definito come:

$$\Delta = b^2 - 4ac.$$

Esercizio 6

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

```
int soluzioni(double a, double b, double c, double *x1, double *x2) {  
    ...  
}
```

La funzione, utilizzando quella per il calcolo del discriminante dell'esercizio precedente e quella per il calcolo della radice quadrata vista a lezione, calcola le soluzioni dell'equazione di secondo grado

$$ax^2 + bx + c = 0$$

e ne ritorna il numero.

Se $\Delta < 0$, la funzione ritorna 0 e non modifica le variabili puntate da x1 e x2.

Se $\Delta = 0$, la funzione ritorna 1, se $\Delta > 0$ la funzione ritorna 2. In entrambi i casi le soluzioni sono date da

$$x_1 = \frac{-b + \sqrt{\Delta}}{2a}$$
$$x_2 = \frac{-b - \sqrt{\Delta}}{2a}$$

e vengono assegnate alle variabili puntate da x1 e x2.

Esercizio 7

Sia dato il seguente file “main.c”:

```
/* inserire qui la funzione doppio() */

int main(void)
{
    double a0, a1, a2, a3, a4, a5, a6;
    double d = 1;

    a0 = d;
    doppio(&d);
    a1 = d;
    doppio(&d);
    a2 = d;
    doppio(&d);
    a3 = d;
    doppio(&d);
    a4 = d;
    doppio(&d);
    a5 = d;
    doppio(&d);
    a6 = d;
}
```

Aggiungere dove indicato la definizione della funzione `doppio()` che raddoppia il valore del parametro. La funzione deve essere fatta **senza modificare in alcun modo la funzione main() fornita**.

Esercizio 8

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

```
double exp (double x){
    ...
}
```

La funzione deve calcolare il valore di e^x utilizzando la seguente equazione:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

Nella formula l'iterazione prosegue fino all'infinito, ma nella pratica potete fare un numero limitato di iterazioni. Iniziate con 10 iterazioni eseguendo il debug e capendo che cosa succede. Poi trovate una soluzione che termini quando il risultato non cambia rispetto all'iterazione precedente, in modo simile a quanto visto a lezione per l'algoritmo della radice quadrata.