

Gli esercizi seguenti devono essere risolti, compilati e testati utilizzando il debugger. Fate progetti diversi per ogni esercizio. Creare una funzione main() che consenta il test.

Esercizio 1

Creare un file “main.c”. Nel file, si realizzi in linguaggio C la funzione corrispondente alla seguente dichiarazione:

```
extern int ismultiple (int x, int y);
```

La funzione riceve due numeri interi x e y e restituisce 1 se x è un multiplo di y, 0 altrimenti. Un esempio di chiamata è il seguente:

```
int main (void) {  
    int x, y, z;  
    x = 6;  
    y = -3;  
  
    z = ismultiple(x, y);  
  
    return 0;  
}
```

In questo caso z varrà 1.

Esercizio 2

Creare un file “main.c”. Nel file, si realizzi in linguaggio C la funzione corrispondente alla seguente dichiarazione:

```
extern double square_dist(double x1, double y1, double x2, double y2);
```

La funzione riceve le coordinate di due punti $P_1(x_1, y_1)$ e $P_2(x_2, y_2)$ e restituisce il quadrato della distanza tra i due punti, ovvero

$$D^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2$$

Esercizio 3

Creare un file “main.c”. Nel file, si realizzi in linguaggio C la funzione corrispondente alla seguente dichiarazione:

```
extern char max_pot10 (unsigned int val);
```

La funzione riceve un valore intero non negativo val e ritorna l'esponente della massima potenza del dieci contenuta in val. Ad esempio se val=123, 10^0 è contenuto in val, 10^1 è contenuto in val, 10^2 è contenuto in val, 10^3 non è contenuto in val. Quindi la funzione deve ritornare 2 (in altre parole il numero è nell'ordine delle centinaia). Se il numero è 0, la funzione deve ritornare -1. Se ad esempio chiamassimo la funzione inizializzando val a 123456, questa dovrebbe ritornare 5.

Esercizio 4

Creare un file “main.c”. Nel file, si realizzi in linguaggio C la funzione corrispondente alla seguente dichiarazione:

```
extern unsigned int fattoriale (unsigned char val);
```

La funzione deve restituire il fattoriale di val. Se non conoscete la definizione di fattoriale, usate Wikipedia.

Qual è il massimo valore di val per cui è possibile calcolare il fattoriale esattamente con questa funzione?

E se usassimo un double per i calcoli e come tipo di ritorno? Sottolineo il calcolare **esattamente**.

Esercizio 5

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

```
extern 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.

Creare inoltre il file `main.c` che contenga la funzione `main()` e richiami la `divisione`.

Sia `main.c`, sia `divisione.c` devono essere aggiunti allo stesso progetto per dire al linker di collegarli assieme. L'aggiunta di `divisione.h` al progetto è inutile perché i file `.h` non vengono compilati, ma è comoda per aprire il file direttamente da Visual Studio.

Esercizio 6

Si realizzi un file “`matematica.c`” contenente tutte le definizioni delle funzioni realizzate negli esercizi precedenti. Realizzare un file “`main.c`”, con le opportune dichiarazioni, che utilizzi tutte le funzioni presenti in “`matematica.c`”.

Sia “`main.c`”, sia “`matematica.c`” devono essere aggiunti allo stesso progetto per dire al linker di collegarli assieme.

Esercizio 7

Si realizzi un file “`matematica.c`” contenente tutte le definizioni delle funzioni realizzate negli esercizi precedenti e un file “`matematica.h`” contenente tutte le corrispondenti dichiarazioni.

Realizzare un file “`main.c`”, con l'opportuno `#include`, che utilizzi tutte le funzioni dichiarate in “`matematica.h`”.

Esercizio 8

Creare un file “`main.c`”. Nel file, si realizzi in linguaggio C la funzione corrispondente alla seguente dichiarazione:

```
extern 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 9

Creare un file “`main.c`”. Nel file, si realizzi in linguaggio C la funzione corrispondente alla seguente dichiarazione:

```
extern unsigned int prossimo_numero_primo(unsigned int x);
```

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