

Funzioni – Esempio

Creare un programma che utilizzando 3 diverse funzioni realizza l'espressione:

$$\left(\left(\underbrace{(n * 120)}_r - 60 \right) + 6 \right) \quad (\text{valutata } n=6)$$

$\underbrace{\hspace{10em}}_q$

$\underbrace{\hspace{15em}}_p$

Ovvero $r(n)=n*120$; $q(n)=r(n)-60$; $p()=q(6)+6$;

Sequenza chiamante:

S.O. → main → P() → Q() → R()

Funzioni – Esempio

```
#include <stdio.h>

int r(int n) { return (n*120); }

int q(int n) { return (r(n)-60); }

int p() { return q(6)+6; }

main() {
    printf("%d\n", p());
}
```

Eseguire con il DEBUG...

Funzioni – Esempio

The screenshot shows the Microsoft Visual C++ IDE with a C program named `gioco.c` open. The program defines three functions: `R`, `Q`, and `P`, and a `main` function. The `main` function calls `P`, which calls `Q`, which calls `R`. The `Call Stack` window is open, showing the current call stack. The `Context` window shows the current function `R` and its parameter `n` with a value of 6. The `Watch` window is empty. The `Output` window shows the loaded DLLs.

```
#include <stdio.h>
int R(int n) { return (n*120); }
int Q(int n) { return (R(n)-60); }
int P() { return Q(6)+6; }
main() {
    printf("%d\n", P());
}
```

Call Stack

- R(int 6) line 3
- Q(int 6) line 5 + 33 bytes
- P() line 7 + 31 bytes
- main() line 11 + 5 bytes
- mainCRTStartup() line 206 + 25 bytes
- KERNEL32! 7c816fd7()

Context: R(int)

Name	Value
n	6

Watch

Name	Value
------	-------

Loaded 'ntdll.dll', no matching symbolic information found.
Loaded 'C:\WINDOWS\system32\kernel32.dll', no matching symbolic information found.

Ready Ln 3, Col 1 REC COL OVR READ

Funzioni – Esercizio 1

Creare una funzione **float square(float x)**. La funzione deve restituire il quadrato del parametro **x**.

Creare un'altra funzione, di nome **float cube(float x)**, che restituisce invece il cubo del valore **x**.

Progettare quindi e codificare un programma che legge un float da tastiera e restituisce il suo quadrato ed il suo cubo. Per calcolare il quadrato ed il cubo si devono utilizzare le due funzioni sopra definite.

Funzioni – Esercizio 2

Si scriva una funzione

```
int somma_potenze(int a, int n);
```

che dati **a** e **n** deve calcolare

$$\sum_{i=1}^n a^i$$

A tal fine si scriva una funzione

```
int potenza(int x, int y);
```

che dati **x** e **y** deve calcolare **x^y** usando come operazione primitiva il prodotto.

Funzioni – Esercizio 3

Codificare in C le funzioni:

int min_to_sec(int a) che considera il parametro a come minuti e restituisce il numero di secondi corrispondente;
int ore_to_sec(int a) che considera il parametro a come ammontare di ore, e restituisca il numero di secondi corrispondente. Si utilizzi la funzione definita precedentemente.

Definire un possibile main che prende in ingresso tre valori interi, rappresentanti ore, minuti e secondi della durata di un CD Audio. Il programma deve stampare il valore corrispondente in secondi. Utilizzare la struct tempo (ore, minuti, secondi), e la funzione di conversione:

int tempo_to_sec(struct tempo t)

Funzioni – Esercizio 4

Si progetti un programma per la geometria piana che definisca una struct punto (float x,y).

Si crei una funzione “distanza”, che dati in ingresso due punti ne calcoli la distanza euclidea.

Si definisca poi una struct puntopolare (float r,t) per le coordinate polari.

Si creino poi le funzioni per la conversione da coordinate rettangolari a polari e viceversa.

Si utilizzi un main() con opportuni input ed output per dimostrare il funzionamento di tutte le funzioni.

Funzioni – Esercizio 5

Codificare in C la funzione

`int ipotenusa(int a, int b)` che, dati i cateti `a` e `b` di un triangolo rettangolo, restituisce il valore dell'ipotenusa.

A tal scopo si utilizzi il Teorema di Pitagora:

$$Ipotenusa = \sqrt{a^2 + b^2}$$

Per calcolare la radice quadrata si utilizzi la funzione di libreria `sqrt(x)`. Per utilizzare quest'ultima si aggiunga l'istruzione `#include <math.h>` in testa al file.

Definire un possibile main che legga da tastiera due valori che rappresentino i cateti di un triangolo rettangolo, e stampi il valore dell'ipotenusa.

Funzioni – Esercizio 6

Codificare in C la funzione

int perimetro(int a, int b, int c) che, dati i lati a,b,c di un triangolo, ne calcola il perimetro.

Codificare in C la funzione

float area(int a, int b, int c)

che restituisce l'area di un triangolo i cui lati misurano a, b, c.

A tal scopo si usi la formula di Erone:

$$Area = \sqrt{p(p-a)(p-b)(p-c)}$$

Dove p è la metà del perimetro. A tal scopo si includa l'header `<math.h>` e si utilizzi la funzione **sqrt(x)**.

Definire un possibile main che prende in ingresso i tre lati di un triangolo e stampa perimetro ed area.

Funzioni – Esercizio 7

Codificare in C la funzione `int primo(int x)` che restituisce:

1 se x è un numero primo

0 altrimenti.

Si utilizzi a tal proposito l'operatore modulo (%).

Si progetti un programma che legge da tastiera un numero N , e stampa a video tutti i numeri primi compresi tra 0 e N .

Funzioni – Esercizio 8

Si scriva un programma che prenda in ingresso i coefficienti a , b , c di un'equazione di secondo grado e che restituisca i risultati dell'equazione.

A tal fine si scriva una funzione

```
int delta_positivo(float a, float b, float c);
```

che dati a , b , c controlli se esiste il delta dell'equazione, e una funzione

```
float delta(float a, float b, float c);
```

che dati a , b , c restituisca il delta dell'equazione.