

**PROBLEMA: Quoziente di due numeri reali qualsiasi  
(SOTTOPROGRAMMI)**

Scrivere un algoritmo che sia in grado di mostrare a video il valore del **quoziente** di **due** numeri **reali** qualsiasi **x** e **y** calcolato prima attraverso la procedura **DivisioneP()**, poi attraverso la funzione **DivisioneF()**.

Esempio:

Se all'interno della procedura `main()` l'utente dovesse inserire i seguenti valori:

$$\mathbf{x = 6 \quad e \quad y = 4}$$

allora:

- dopo la chiamata al sottoprogramma di tipo **PROCEDURA** **DivisioneP()** dovrò ottenere **1.5**;
- dopo la chiamata al sottoprogramma di tipo **FUNZIONE** **DivisioneF()** dovrò ottenere ugualmente **1.5**.

N.B. In una divisione occorre stare attenti principalmente al valore del divisore (ossia la variabile **y**) che, per ovvie ragioni, non potrà mai assumere il valore 0

## TABELLE DEI DATI

<b>DATI DI INPUT DEL PROBLEMA PRINCIPALE PROCEDURA main()</b>				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
<b>x</b>	REAL	STATICA	TUTTI	Dividendo acquisito da tastiera
<b>y</b>	REAL	STATICA	<b>y ≠ 0</b>	Divisore acquisito da tastiera

<b>DATI DI OUTPUT DEL PROBLEMA PRINCIPALE PROCEDURA main()</b>				
Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
<b>quoz</b>	REAL	STATICA	TUTTI	Risultato del quoziente di <b>x</b> e di <b>y</b> mostrato a video, calcolato prima con la procedura <b>DivisioneP()</b> e poi con la funzione <b>DivisioneF()</b>

<b>DATI DI ELABORAZIONE o DI LAVORO DEL PROBLEMA PRINCIPALE PROCEDURA main()</b>				
Nome variabile oppure nome costante	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione

---

**DATI DI INPUT DEL SOTTOPROBLEMA: PROCEDURA DivisioneP()**

Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
<b>a</b>	INT	STATICA	TUTTI	Primo parametro FORMALE, PASSATO PER VALORE, che conterrà il valore del DIVIDENDO
<b>b</b>	INT	STATICA	<b>b ≠ 0</b>	Secondo parametro FORMALE, PASSATO PER VALORE, che conterrà il valore del DIVISORE supposto NON NULLO

**DATI DI OUTPUT DEL SOTTOPROBLEMA: PROCEDURA DivisioneP()**

Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
<b>q</b>	INT	STATICA	TUTTI	Terzo parametro FORMALE, PASSATO PER RIFERIMENTO, che conterrà il valore del quoziente della divisione tra DIVIDENDO e DIVISORE

**DATI DI ELABORAZIONE o DI LAVORO DEL SOTTOPROBLEMA: PROCEDURA DivisioneP()**

Nome variabile oppure nome costante	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione

**DATI DI INPUT DEL SOTTOPROBLEMA: FUNZIONE DivisioneF()**

Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
<b>a</b>	INT	STATICA	TUTTI	Primo parametro FORMALE, PASSATO PER VALORE, che conterrà il valore del DIVIDENDO
<b>b</b>	INT	STATICA	<b>b ≠ 0</b>	Secondo parametro FORMALE, PASSATO PER VALORE, che conterrà il valore del DIVISORE supposto NON NULLO

**DATI DI OUTPUT DEL SOTTOPROBLEMA: FUNZIONE DivisioneF()**

Nome variabile	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione

**DATI DI ELABORAZIONE o DI LAVORO DEL SOTTOPROBLEMA: FUNZIONE DivisioneF()**

Nome variabile oppure nome costante	Tipo dati	Tipo Allocazione	Valori ammessi	Descrizione
<b>q</b>	INT	STATICA	TUTTI	Variabile che conterrà il valore del quoziente della divisione tra DIVIDENDO e DIVISORE da restituire nel nome della funzione

**ALGORITMO** Quoziente\_2\_Reali\_SOTTOGRAMMA**PROCEDURA** main ( )x, y, quoz : **REAL****INIZIO**

Leggi(x)

**RIPETI**

Leggi(y)

**FINCHE'** (y ≠ 0)

/\* Chiamata alla PROCEDURA DivisioneP() \*/

**DivisioneP** (x, y, quoz)

Scrivi(quoz)

/\* Chiamata alla FUNZIONE DivisioneF() \*/

quoz ← **DivisioneF**(x, y)

Scrivi(quoz)

**RITORNA****FINE****PROCEDURA** **DivisioneP** (VAL a : **REAL**, VAL b : **REAL**, REF q : **REAL**)**INIZIO**

q ← a / b

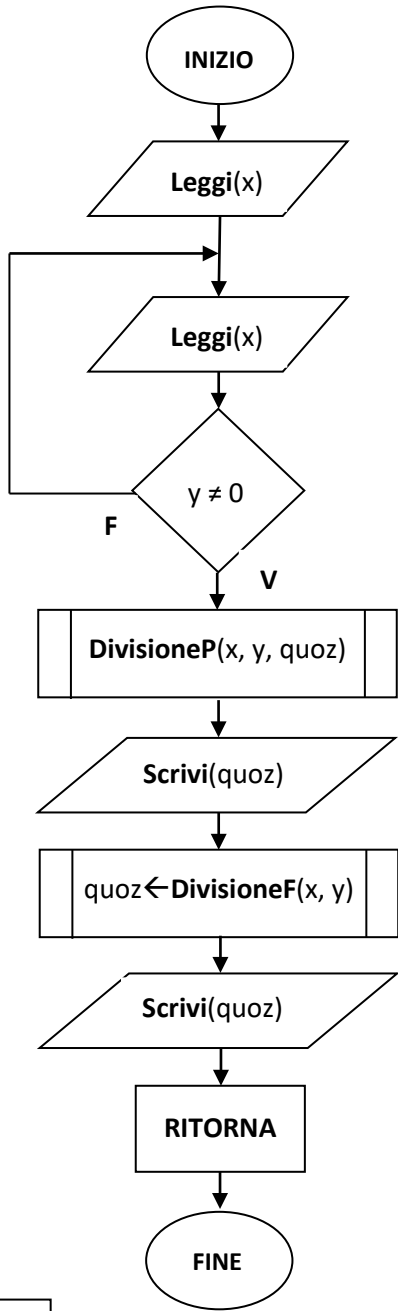
**RITORNA****FINE****FUNZIONE** **DivisioneF** (VAL a : **REAL**, VAL b : **REAL**) : **REAL**q : **REAL****INIZIO**

q ← a / b

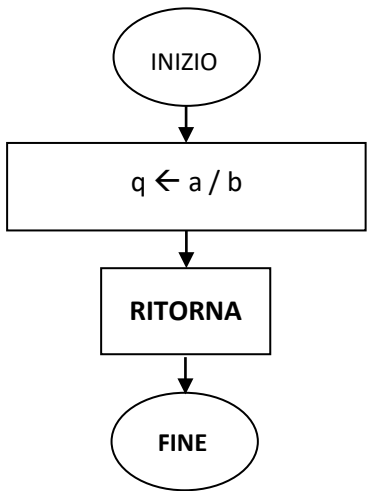
**RITORNA** (q)**FINE**

**PROCEDURA main()**

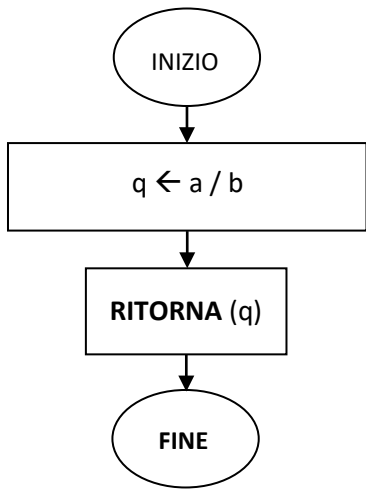
**FLOWCHART**



**PROCEDURA *DivisioneP()***



**FUNZIONE *DivisioneF()***



## CODIFICA C

```
#include <stdio.h>
#include <stdlib.h>

// N.B. ESCLUSIVAMENTE nei prototipi può essere omissa il NOME DEI PARAMETRI FORMALI, NON IL TIPO
// definizione dei prototipi delle funzione DivisioneP (teoricamente è una PROCEDURA)
void DivisioneP(float a, float b, float*q);
// definizione dei prototipi delle funzione DivisioneF (teoricamente è una FUNZIONE)
float DivisioneF(float a, float b);

////////////////////////////////////
//     FUNZIONE main()           //
////////////////////////////////////
int main(int argc, char*argv[])
{
    /* dati di input */
    float x, y;

    /* dati di output */
    float quoz;

    printf("Inserisci il DIVIDENDO: ");
    scanf("%f", &x);

    do
    {
        printf("Inserisci il DIVISORE (non nullo: ");
        scanf("%f", &y);
    }
    while (y == 0);

    //CALL alla funzione DividiP() (teoricamente è una PROCEDURA)
    DivisioneP(x, y, &quoz);
    printf("Il quoziente calcolato con la PROCEDURA DividiP() e': %.2f\n", quoz);

    //CALL alla funzione DividiF() (teoricamente è una FUNZIONE)
    quoz = DivisioneF(x, y);
    printf("Il quoziente calcolato con la FUNZIONE DividiF() e': %.2f\n", quoz);

    return 0;
}

////////////////////////////////////
//     FUNZIONE DivisioneP()     // // (teoricamente è una PROCEDURA)
////////////////////////////////////
void DivisioneP(float a, float b, float*q)
{
    *q = a / b;
    return;
}

////////////////////////////////////
//     FUNZIONE DivisioneF()     // // (teoricamente è una FUNZIONE)
////////////////////////////////////
float DivisioneF(float a, float b)
{
    //dati di lavoro e di output
    float q;

    q = a / b;
    return (q);
}
```