

Laboratorio di informatica

Ingegneria meccanica

*Esercitazione 8 - 21 Novembre
2007*

1

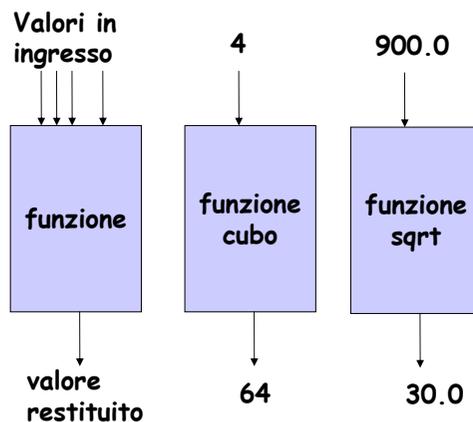
Funzioni in C

- Un programma C si compone di **funzioni**
- E' disponibile una collezione *predefinita* di funzioni che possono essere usate *direttamente* in ogni programma (**libreria standard**)
- Per introdurre una **nuova funzione** occorre specificare:
 - la sequenza di **operazioni** da eseguire
 - gli **argomenti** su cui operare
 - il **risultato** da restituire
- Un programma C viene modularizzato con l'uso di funzioni

2

Funzioni come scatola nera

A partire da uno o più valori in ingresso le funzioni restituiscono un valore al modulo chiamante



3

C: Programma con funzioni

- Dichiarare le funzioni tramite i prototipi prima del main
- Definire le funzioni corrispondenti ai prototipi dopo il main
- Inserire le chiamate delle funzioni dove necessario

4

Esempio di funzione C

```
...
int cubo( int ); /* prototipo funzione cubo */

int main( )
{
    int num ;
    ... /* attivazione funzione cubo */
    printf( "cubo(%d) = %d", num , cubo( num ) );
    ...
    return 0;
}

/* definizione funzione cubo */
int cubo( int n )
{
    return n*n*n ;
}
```

5

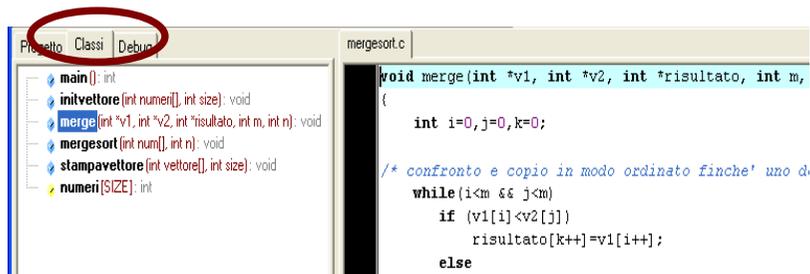
C: Passaggio degli argomenti

- Gli argomenti presenti nel prototipo e nella definizione di una funzione vengono detti anche **argomenti** (o parametri) **formali**, distinguendoli così dagli argomenti effettivamente utilizzati durante la chiamata della funzione, detti **argomenti** (o parametri) **attuali**
- Le espressioni passate come parametri attuali vengono valutate, convertite ai tipi dichiarati nel prototipo e assegnate ai rispettivi parametri formali
- Le variabili coinvolte nell'espressione sono passate per **valore** e non possono essere modificate dalla funzione

6

DevCpp: funzioni

E' possibile vedere le funzioni utilizzate nel programma. Per ogni funzione viene indicato nome, parametri e tipo del valore restituito



Andare alla scheda "Classi"

7

Esercizio 1

Scrivere un programma che, acquisiti da stdin 3 valori reali (float) **a**, **b**, **c**, usi due funzioni per determinare, rispettivamente, minimo e massimo tra i valori acquisiti, e visualizzi su standard output i risultati restituiti dalle funzioni

8

Esercizio 1: suggerimenti

- Prototipo funzione
`float valoreMin (float , float , float) ;`
- Definizione funzione
`float valoreMin (float v1, float v2, float v3)
{
 return min ;
}`
- Attivazione funzione
`minimo = valoreMin (a , b , c) ;`

9

Esercizio 2

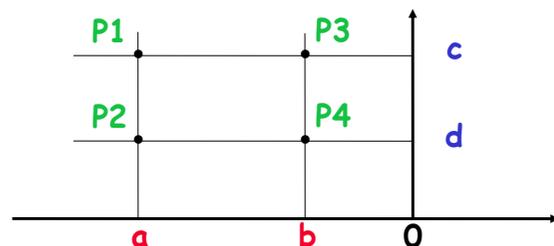
Progettare una funzione che, ricevuti un carattere (char) `c` e due valori reali (float) `x` e `y`, restituisca il reale (float) corrispondente all'operazione specificata da `c` ('+' sta per somma, '-' sta per differenza, '*' sta per prodotto, '/' sta per divisione) eseguita tra `x` e `y`. In caso di operatore non valido (carattere `c` diverso dai quattro valori specificati sopra), la funzione restituirà il valore `0.0` e non effettuerà alcuna operazione tra `x` e `y`.

Inserire poi la funzione in un programma di prova, che provveda all'acquisizione da standard input degli argomenti della funzione, alla chiamata della funzione e alla visualizzazione su standard output del valore da questa restituito

10

Esercizio 3

Scrivere un programma che, acquisiti da stdin 4 valori reali `a`, `b`, `c`, `d`, usi una funzione per il calcolo dell'area del rettangolo di punti `P1`, `P2`, `P3`, `P4`, di coordinate `(a, c)`, `(a, d)`, `(b, c)`, `(b, d)`, e stampi poi su stdout il valore restituito dalla funzione



Esempio con `a` e `b` negativi e `c` e `d` positivi

11

Esercizio 3: suggerimenti

- Prototipo funzione
`float areaRect(float , float , float , float) ;`
- Definizione funzione
`float areaRect(float a , float b , float c , float d)
{ ...
 return area ;
}`
- Attivazione funzione
`arearettangolo = areaRect(ar , br , cr , dr) ;`

12

Esercizio 4: somma di due frazioni

Scrivere un programma per **calcolare la somma di due frazioni**.

Il programma deve:

1. ricevere in input quattro interi (num1, den1, num2, den2) corrispondenti ai numeratori e ai denominatori delle frazioni da sommare
2. fornire in output numeratore e denominatore della frazione (eventualmente semplificata) risultante dalla somma dei due numeri.

13

Esercizio 4: funzioni del programma

- funzione **massimoComunDivisore** che calcola il massimo comun divisore di due interi
- funzione **minimoComuneMultiplo** che calcola il minimo comune multiplo di due interi
- funzione **main**

14

Esercizio 4: massimoComunDivisore

Calcolare il massimo comune divisore tra due numeri **a** e **b** in ingresso.

- **Algoritmo di Euclide**: permette di determinare il M.C.D. di due numeri naturali in modo molto più efficiente che non elencando tutti i divisori dei due numeri considerati.

```
do {  
    r = a % b;  
    a = b;  
    b = r;  
} while (r != 0)
```

Esempio a = 12 b=15

1) r=12 a= 15 b=12
2) r=3 a=12 b=3
• r=0 a=3 b=0

MCD = 3

Il massimo comun divisore è **a**

15

Esercizio 4: minimoComuneMultiplo

- La funzione **minimoComuneMultiplo** utilizza la funzione **massimoComunDivisore**
- Il minimo comune multiplo è uguale al prodotto dei due numeri diviso il loro massimo comun divisore,
 $mcm(a,b) = a*b / MCD(a,b)$

16

Esercizio 4: main

La funzione **main**:

- legge l'input
- calcola il risultato usando gli altri moduli
- verifica se il risultato deve essere semplificato
- stampa il risultato

Per effettuare la somma di due frazioni si dovrà

1. calcolare il mcm dei due denominatori che costituisce il denominatore della frazione_somma
2. calcolare i numeratori delle frazioni equivalenti a quelle date con il nuovo denominatore
3. sommare i nuovi numeratori per avere il numeratore della frazione_somma

Per semplificare una frazione si dovrà calcolare il MCD tra numeratore e denominatore

17

Esercizio 5

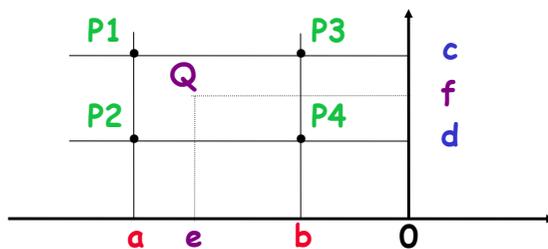
Scrivere un programma che, acquisiti da stdin 6 valori reali (float) **a, b, c, d, e, f**, usi una funzione per verificare se il punto **Q** di coordinate **(e, f)** è interno o esterno al rettangolo di punti **P1, P2, P3, P4**, di rispettive coordinate **(a, c), (a, d), (b, c), (b, d)**, e visualizzi su stdout l'esito della verifica. La funzione restituisce un intero di valore **1/0** se il punto **Q** è **interno/esterno** al rettangolo.

Si considerino esterni i punti che si trovano sul perimetro del rettangolo.

18

Esercizio 5: esempio

Esempio con **a** e **b** negativi, **c** e **d** positivi e punto **Q** interno al rettangolo



19