

FONDAMENTI DI INFORMATICA

Prof. PIER LUCA MONTESSORO
Università degli Studi di Udine

Linguaggio C
Esercizi sulle funzioni



Volume dei solidi

Si scrivano le funzioni:

`vsfera, vcubo,`

`vparall_ret_bas_triang`

che restituiscono la misura del volume
rispettivamente di una sfera, di un cubo e di un
parallelepipedo retto a base triangolare.



Volume dei solidi

Si scriva un programma che legga da tastiera le misure di base di un certo numero di solidi (sfere, cilindri e parallelepipedi retti a base triangolare) e stampi il volume totale.

segue esempio ...



Volume dei solidi

```
C:\PROGRAMMI> volumi
quanti solidi? 4
tipo di solido n. 1 (s, c, p): s
- raggio: 2.3
tipo di solido n. 2 (s, c, p): c
- lato: 4.0
tipo di solido n. 3 (s, c, p): p
- base del triangolo di base: 5.1
- altezza del triangolo di base: 3.2
- altezza del parallelepipedo: 1.3
tipo di solido n. 4 (s, c, p): s
- raggio: 1.7
VOLUME TOTALE: 146.152537
C:\PROGRAMMI>
```



Controllo della data

Si scriva la funzione

`checkdate (int g, int m, int a)`

che controlla la validità della data passata come parametro.

Si scriva anche un main per provare tale funzione.



Controllo della data

- Esempi:

```
C:\PROGRAMMI> chkdate
```

```
Inserire la data (g m a): 29 2 2000  
data corretta
```

```
C:\PROGRAMMI> chkdate
```

```
Inserire la data (g m a): 30 2 2000  
data errata
```

```
C:\PROGRAMMI> chkdate
```

```
Inserire la data (g m a): 29 2 2001  
data errata
```



Calcolo di $y = \sin(x)$

- Si scriva un programma C che calcoli e stampi la funzione $\sin(x)$, con x letto da tastiera, utilizzando lo sviluppo in serie:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Il calcolo della funzione deve terminare quando si è raggiunta la precisione voluta (parametro letto da tastiera)



Lettura di un numero intero esadecimale

- Si scriva la funzione

`int leggi_hex(void)`

che legge da tastiera un numero intero positivo in base 16 e lo restituisce come valore di ritorno.

- Il numero è introdotto cifra per cifra e letto con la funzione `getchar()`, e quindi ogni operazione di input restituisce il codice ASCII di una cifra del numero, ovviamente a partire da quella più significativa. Il numero termina con il codice ASCII “carriage return” (`'\n'`).



MCD

- Si scrivano due versioni della funzione

```
int mcd (int a, int b)
```

che calcola il massimo comun divisore tra due numeri interi passati come argomenti.

- Le due versioni devono utilizzare due diversi algoritmi:
 - decrementi successivi
 - metodo di Euclide



MCD (m, n) mediante decrementi successivi

MCD = minimo tra m ed n

controlla se m e n sono entrambi
divisibili per MCD:

se sì, stampa e termina;

se no, decrementa di 1 MCD e ripeti
(finché MCD non diventa 1)



MCD (m, n) tramite il metodo di Euclide

```
n1 = n;  
m1 = m;  
while (n1 è diverso da m1)  
{  
    sottrai dal maggiore tra n1 e m1  
    l'altro numero  
}
```

alla fine del ciclo n1 (o m1, avendo lo stesso valore) è il MCD



Radice quadrata con il metodo di Newton

- Si scriva la funzione

```
double sqrt_Newton (double x)
```

che calcola la radice quadrata di x tramite il metodo di Newton

```
n1 = n;  /* n è il radicando */  
n2 = 1;  
while (|n1 - n2| è maggiore  
        della precisione cercata)  
{  
    n1 = (n1 + n2) / 2;  
    n2 = n / n1;  
}
```



Fattoriale

- Si scriva un programma per provare la funzione fattoriale ricorsiva con le stampa necessarie a seguire la traccia dell'esecuzione (ingresso nella funzione, calcoli, fine e valore di ritorno, ecc.)



Fibonacci

- Si scriva una versione ricorsiva della successione di Fibonacci:

$$F_0 = 0$$

$$F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2}$$

