

# **ESERCITAZIONE N.1**

## **LINGUAGGIO C**

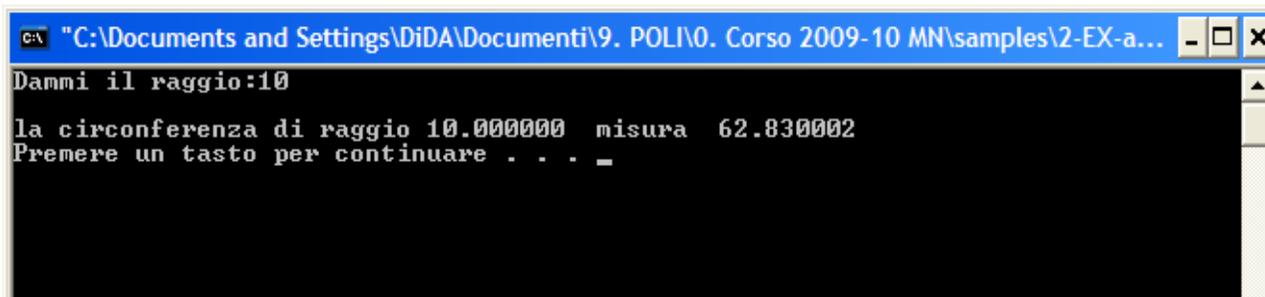
**In preparazione alla prova in Itinere**



# DATO IL RAGGIO TROVARE LA CIRCONFERENZA

circonfenza.c

```
1 #include <stdio.h>
2 #define pigreco 3.1415
3 main()
4 {
5 float raggio;
6 float circonferenza;
7
8 printf("Dammi il raggio:");
9 scanf("%f",&raggio);
10
11 circonferenza = 2* pigreco * raggio;
12
13 printf("\nla circonferenza di raggio %f misura %f\n",raggio, circonferenza);
14 }
```



```
C:\Documents and Settings\DiDA\Documenti\9. POLI\0. Corso 2009-10 MN\samples\2-EX-a... - □ ×
Dammi il raggio:10
la circonferenza di raggio 10.000000 misura 62.830002
Premere un tasto per continuare . . . _
```



# PROGRAMMA CHE LEGGE N NUMERI IN UN VETTORE E LI VISUALIZZA IN ORDINE INVERSO

```
stampainverso.c
1 /* Programma che legge n numeri in un vettore e li visualizza in ordine inverso */
2 #include <stdio.h>
3 #define MAX 100
4 int main (){
5     int n, i, numeri[MAX ];
6     /* Lettura dimensione vettore */
7     printf ("Inserire il numero di elementi:");
8     scanf ("%d", &n);
9     if (n>MAX) printf ("Valore troppo elevato\n");
10    else{
11        /* lettura dei numeri */
12        for (i=0; i<n; i++){
13            printf ("Inserire un numero:");
14            scanf ("%d", &numeri[i]);
15        }
16        /* visualizzazione dei numeri in ordine inverso */
17        for(i=n-1; i>=0; i--) printf ("%d\n", numeri[i]);
18    }
19    return 0;}
20
```

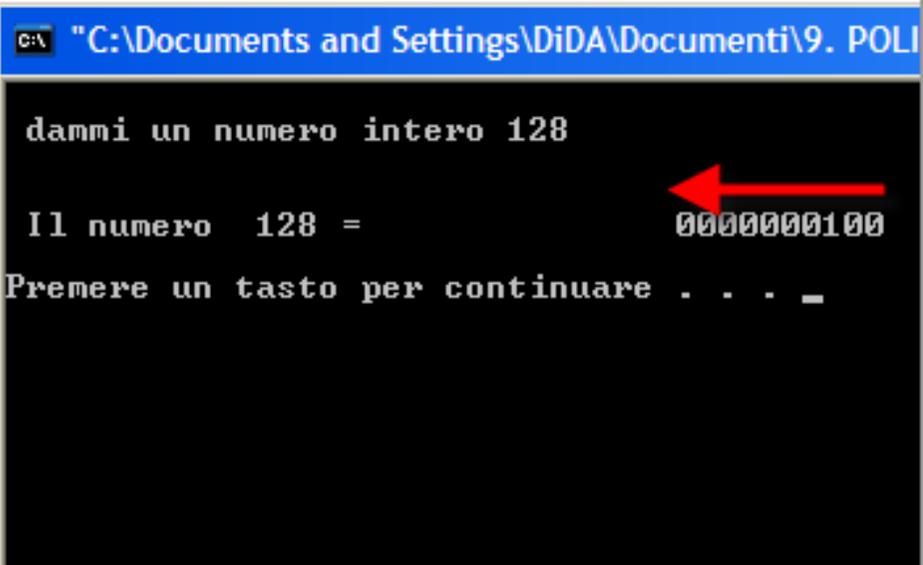
```
C:\Documents and Settings\DiDa\Documents\9. POLI\0. Corso 2009-10 MN\samples\6-EX-a...
Inserire il numero di elementi:6
Inserire un numero:1
Inserire un numero:5
Inserire un numero:3
Inserire un numero:7
Inserire un numero:2
Inserire un numero:3
3
2
7
3
5
1
Premere un tasto per continuare . . .
```

Configuration:  
Checking file dependency...  
Compiling C:\Documents and...  
Linking...



# DATO UN INTERO STAMPARNE LA RAPPRESENTAZIONE BINARIA

```
2f3-graph.c conversione.c
1 #include <stdio.h>
2 int main(int argc, char *argv[])
3 {
4     int hhh, I;
5     printf("\n dammi un numero intero\t");
6     scanf("%d",&hhh);
7     printf("\n\n Il numero  %d =\t\t",hhh);
8
9     /* converte in 10 cifre binarie */
10    for(I=0;I<10 ;I++ )
11    {
12        printf("%d",hhh%2);
13
14        hhh=hhh/2;
15    }
16    printf("\n\n");
17
18    return 0;
19 }
```



```
C:\Documents and Settings\DiDA\Documenti\9. POLI
dammi un numero intero 128

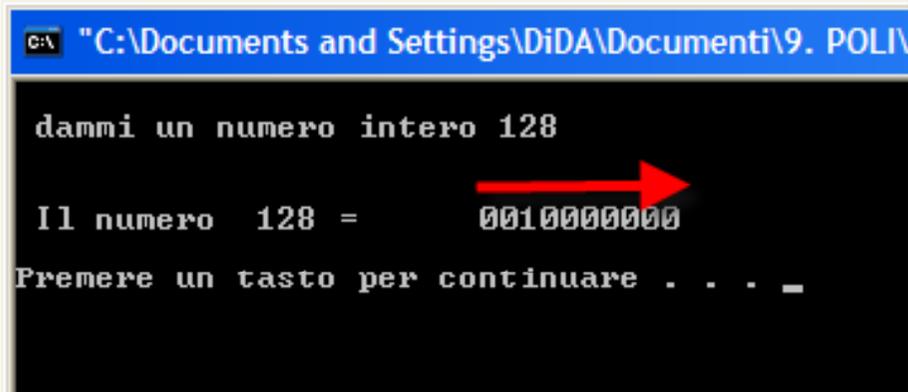
Il numero 128 = 0000000100
Premere un tasto per continuare . . . _
```



# DATO UN INTERO STAMPARNE LA RAPPRESENTAZIONE BINARIA (ALTERNATIVA)

f2f3-graph.c conversione-alt.c

```
1 #include <stdio.h>
2 int main(int argc, char *argv[])
3 {
4     int hhh, I;
5     printf("\n dammi un numero intero\t");
6     scanf("%d",&hhh);
7     printf("\n\n Il numero  %d =\t\t",hhh);
8
9     /* converte in 10 cifre binarie */
10    for(I=0;I<10 ;I++ ) stampa 1 carattere
11    { torna indietro di 2 car
12    printf("%d\b\b",hhh%2);
13
14    hhh=hhh/2;
15    }
16    printf("\n\n");
17
18    return 0;
19 }
```



```
C:\Documents and Settings\DiDA\Documenti\9. POLI...
dammi un numero intero 128
Il numero 128 = 0010000000
Premere un tasto per continuare . . . _
```



# DATO UN INTERO STAMPARNE LA RAPPRESENTAZIONE BINARIA (SOLUZIONE)

```
2f3-graph.c | conversione-soluzione.c
1 #include <stdio.h>
2 #define NCIFRE 15
3 int main(int argc, char *argv[])
4 { /*stampa un numero decimale in base due su NCIFRE cifre binarie*/
5     int hhh, I;
6     int num_conv[NCIFRE];
7     printf("\n dammi un numero intero\t");
8     scanf("%d",&hhh);
9     printf("\n\n Il numero  %d =\t\t",hhh);
10
11 /* converte in NCIFRE cifre binarie */
12 for(I=0;I<NCIFRE ;I++)
13 {
14     num_conv[I]=hhh%2;
15     hhh=hhh/2;
16 }
17 for(I=0; I<NCIFRE; I++)
18 {printf("%d",num_conv[(NCIFRE-1)-I])
19 }
20 printf("\n\n");
21
22 return 0;

```

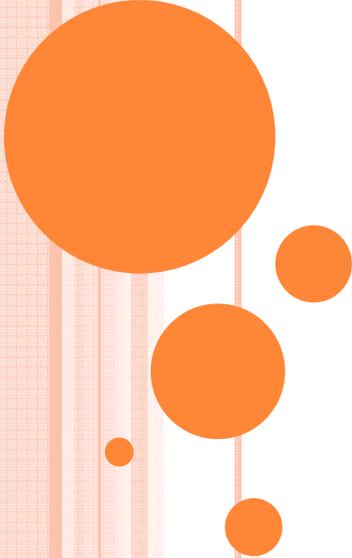
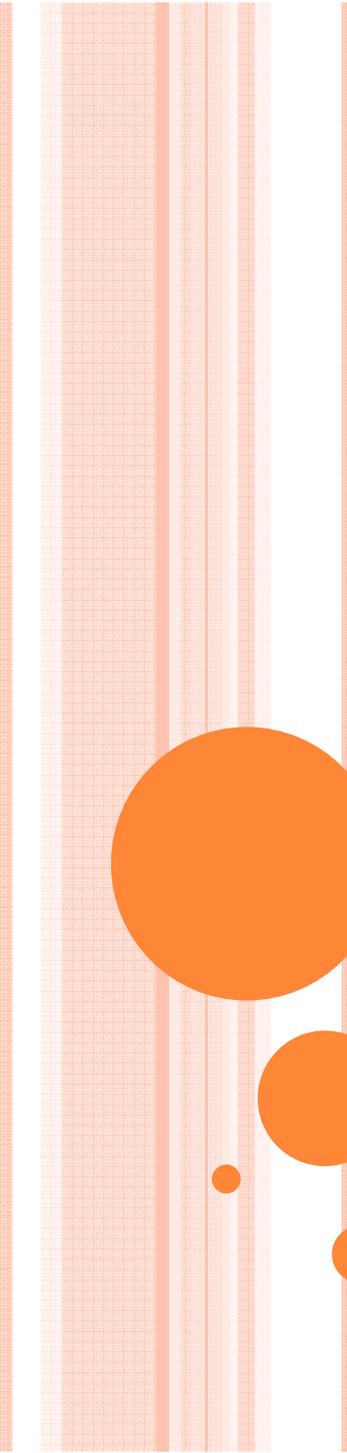
Configuration: minaw2.95 - CUI Debug. Builder Tvr

```
C:\Documents and Settings\DiDA\Documenti\9. POLI\0. Co
dammi un numero intero 2047

Il numero  2047 =          0000111111111111
Premere un tasto per continuare . . .

inserisce le cifre da sx
le stampa da dx
```





**CALCOLARE LA RADICE  
QUADRATA DI UN NUMERO REALE  
POSITIVO**



# METODI PER IL CALCOLO DELLA RADICE QUADRATA

- **Note storiche**
- I primi ad occuparsi del problema dell'estrazione di radice quadrata di un numero sono stati i babilonesi.
- Essi, tra i primi ad utilizzare un sistema di numerazione posizionale, avevano elaborato un procedimento per l'estrazione di radice quadrata.
- I babilonesi avevano ricavato un valore di pari a 1,414222 con un errore di circa 0,000008 dal valore vero.
- 



## METODO BABILONESE

- Un algoritmo per approssimare  $\sqrt{z}$  comunemente usato è conosciuto come metodo babilonese e sfrutta gli stessi principi poi codificati nel metodo di Newton.
- **Qu**
  1. Poni  $n=1$  e inizia con un valore arbitrario positivo  $x_n$  (quanto più esso è prossimo alla radice, tanto migliore è la convergenza dell'algoritmo)
  2. sostituisci  $x_n$  con la media di  $x_n$  e  $z/x_n$
  3. aumenta  $n$  e vai al punto 2

Questo è un algoritmo a convergenza quadratica, il che vuol dire che il numero di cifre corrette  $r$  grossolanamente raddoppia ad ogni passo.

Questo algoritmo può essere rappresentato dalla relazione di ricorrenza

$$x_{n+1} = 0,5 \left( x_n + \frac{z}{x_n} \right)$$

dalla quale si ricava  $\lim_{n \rightarrow \infty} x_n = \sqrt{z}$ .



## ESEMPIO

Ad esempio, poiché la radice quadrata di 2 deve essere compresa tra 1 e 2, stimiamo che sia circa 1,5.

Applicando ripetutamente la formula otteniamo i seguenti valori:

$$x_0 = 1,5$$

$$x_1 = \frac{1,5 + 2/1,5}{2} = 1,416667$$

$$x_2 = \frac{1,416667 + 2/1,416667}{2} = 1,414216$$

$$x_3 = \frac{1,414216 + 2/1,414216}{2} = 1,414214$$

$$x_4 = \frac{1,414214 + 2/1,414214}{2} = 1,414214$$

...

radice.c

```
1 #include <stdio.h>
2 int main(int argc, char *argv[])
3 {float dato, approx;
4 const float epsilon = 0.00001;
5
6 printf("Inserisci un numero reale >0.0 \t");
7 scanf("%f",&dato );
8
9 if(dato<=0.0) /*controllo il dato di partenza */
10     {printf("\n\n dato di partenza errato\t  %f\n",dato);
11     return 1;
12     };
13 approx=(1+dato)*0.5; /*prima approssimazione = media tra 1 e il dato */
14
15 while( (approx*approx - dato ) > epsilon )
16 {
17     printf("\n\t radice quadrata = %f\n", approx);
18     approx=0.5* (approx + dato/approx); /*formula ricorrente*/
19 }
20
21 printf("\n\t radice quadrata = %f\n", approx);
22
23     return 0;
24 }
25 }
```

Conf  
Checking file depen

```
C:\ "C:\Programmi\C-Free Standard\samples\5-EX-while\radice.exe"
Inserisci un numero reale >0.0 2.0
radice quadrata = 1.500000
radice quadrata = 1.416667
radice quadrata = 1.414216
Premere un tasto per continuare . . . _
```