

## LABORATORIO DI PROGRAMMAZIONE E CALCOLO

Canale 1 (A-K) A.A. 2024/2025

Foglio di esercizi n. 4

1. Scrivere un programma C++ che acquisisce da tastiera i coefficienti reali  $a$ ,  $b$ ,  $c$  di una equazione di secondo grado  $ax^2 + bx + c = 0$ , controlla che  $a$  sia diverso da “zero” (altrimenti termina con un messaggio di errore), calcola le radici reali e distinte dell’equazione (con un algoritmo stabile) e le stampa sul video, oppure segnala che non ci sono radici reali e distinte.

```
/* Esercizio F4_1 ... */
#include <stdio.h>
#include <math.h>
#define TOL 1.e-8

int main()
{double a,b,c,delta,x1,x2;
printf( "Primo coefficiente:      a = ");
scanf("%lf",&a);
if (a>-TOL && a<TOL)
    {printf("L'equazione data non e' di secondo grado \n");
    return 1;}
printf( "Secondo coefficiente:   b = ");
scanf("%lf",&b);
printf( "Terzo coefficiente:     c = ");
scanf("%lf",&c);
delta=b*b-4*a*c;
if (delta<TOL)
    printf("Non ci sono due radici reali distinte\n");
else if (b<0)
    {delta=sqrt(delta);
    x1=(-b+delta)/(2*a);
    x2=2*c/(-b+delta);
    printf("Prima radice      = %e\n",x1);
    printf("Seconda radice   = %e\n",x2);}
else
    {delta=sqrt(delta);
    x1=2*c/(-b-delta);
    x2=(-b-delta)/(2*a);
    printf("Prima radice      = %e\n",x1);
    printf("Seconda radice   = %e\n",x2);}
return 0;
}
```

2. Scrivere un programma C++ che implementa l'algoritmo di Archimede (sia nella versione originale che nella versione stabile), stampa sul video - visualizzandone sedici cifre significative - le approssimazioni di  $\pi$  ottenute dal calcolo del semiperimetro dei poligoni regolari di  $2^{k+2}$  lati inscritti nel cerchio di raggio unitario, con  $0 \leq k \leq 25$ , e confronta l'ultima stima per difetto ottenuta sia con la stima per eccesso data dal semiperimetro del corrispondente poligono regolare circoscritto al cerchio di raggio unitario che con la costante 'M\_PI'.
3. Scrivere un programma C++ che acquisisce da tastiera un numero reale  $x$  e un numero intero  $n$ . Se  $n$  non è negativo, calcola e stampa sul video  $x^n$ , altrimenti (ovvero se  $n$  è negativo), stampa sul video *inf* nel caso che  $x$  è "zero", o calcola e stampa sul video  $1/x^{-n}$  altrimenti (ovvero nel caso che  $x$  è diverso da "zero"). In tutti i casi si confronta con l'output dato dalla funzione 'pow' chiamata con argomenti  $x$  e  $n$ .
4. Scrivere un programma C++ che calcola e stampa sul video la seguente approssimazione del numero di Nepero:

$$e = \sum_{k=0}^{16} \frac{1}{k!}$$

e ne stampa sul video l'errore relativo facendo uso della costante 'M\_E'.