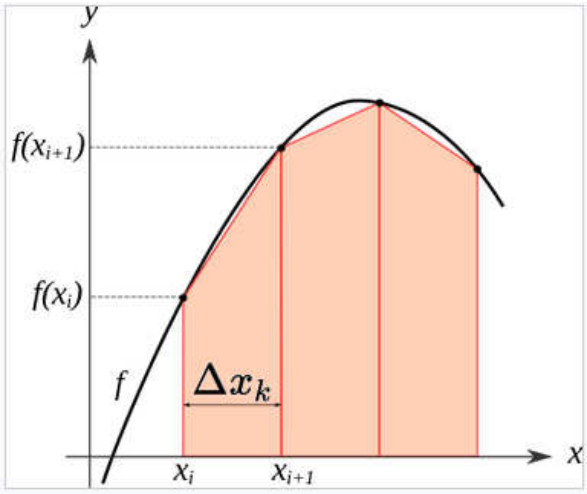
Programmazione C++

Esercizio 0

Si vuole scrivere un programma che calcola l'integrale di una funzione *f(x)* utilizzando la tecnica di approssimazione dei trapezi. In pratica

$$\int_a^b f(x)\,dx \approx \sum_{k=1}^{N} \frac{f(x_{k-1}) + f(x_k)}{2} \Delta x_k$$



dove $[a,b]$ è l'intervallo di integrazione, N è il numero di sotto-intervalli in cui è suddiviso $[a,b]$, e Δx_k è la dimensione dei sotto-intervalli.

In un file `main.cpp`, scrivere il codice per calcolare l'integrale della funzione `sin(x)` nell'intervallo $[0,\pi]$ chiedendo all'utente il numero di sotto-intervalli da utilizzare per l'approssimazione. Stampare a video il valore ottenuto.

Programmazione C++

Esercizio 1

Il metodo di Newton per la soluzione di equazioni non lineari può essere sfruttato per il calcolo della radice quadrata. Dato un numero reale positivo x e una tolleranza $\epsilon > 0$, il metodo esegue i seguenti passi:

1. si sceglie una stima iniziale s_0 (es: $s_0 = \frac{x}{2}$);
2. si migliora la stima: $s_k = \frac{s_{k-1}^2 + x}{2 s_{k-1}}$;
3. se $|s_k^2 - x| < \epsilon$ il metodo termina, altrimenti si ripetono i passi 2 e 3.

L'ultimo s_k calcolato corrisponde ad una stima di $\sqrt{x}$.

Creare i file `nsqrt.h` e `nsqrt.cpp` che dichiarino e definiscano la funzione

```
double nsqrt(double x, double epsilon)
```

che implementa il metodo descritto.

Si scriva quindi il file `test_nsqrt.cpp` che definisca la funzione `main` che: (i) legga x dallo standard input; (ii) scriva sullo standard output l'approssimazione della sua radice quadrata e il valore restituito dalla funzione di libreria `std::sqrt` dichiarata nel file header `cmath`.

Esercizio 2

Il sistema di numerazione romano è un sistema di numerazione additivo, ovvero a ogni simbolo è associato un valore e il numero rappresentato è dato dalla somma dei valori dei simboli. I simboli ed i rispettivi valori sono: I = 1, V = 5, X = 10, L = 50, C = 100, D = 500 e M = 1000. Valgono le seguenti regole:

- i simboli I, X, C e M possono essere ripetuti al massimo tre volte;
- i simboli V, L e D non possono essere mai inseriti più di una volta;

- una sequenza (ovvero una stringa) di simboli che non presenta mai valori crescenti denota l'intero ottenuto sommando i valori dei simboli indicati; esempi II = 2, XI = 11, XVIII = 18, CXV = 115, DLII = 552, MMVII = 2007.
- Quando si incontra un simbolo seguito da un secondo simbolo di valore maggiore si ha come risultato la differenza tra i due: IV = 4, IX = 9, XL = 40, XC = 90, CD = 400, CM = 900.
- Solo I, X e C possono essere usati in senso sottrattivo.
- Solo i numeri interi compresi tra 1 e 3999 possono essere rappresentati in questa notazione.

L'esercizio consiste nell'implementazione della funzione

```
void print_roman(int n)
```

che stampa sullo standard output il valore dell'argomento `n` nel sistema di numerazione romano. La funzione dovrà essere dichiarata nel file `roman.h` e definita nel file `roman.cpp`.

Creare quindi il file `test_roman.cpp` contenente la funzione `main` che (i) legge dallo standard input un numero intero; (ii) chiama la funzione `print_roman`. Prima di chiamare la funzione bisogna controllare che il valore sia rappresentabile come numero romano e, in caso contrario, si stampi un messaggio di errore sullo standard error.

È possibile controllare la correttezza del programma tramite il sito `http://www.novaroma.org/via_romana/numbers.html`.

Esercizio 3

Creare i file `date.h` e `date.cpp` che dichiarino e definiscano la funzione

```
int date_difference(int year1, int month1, int day1,
                    int year2, int month2, int day2)
```

che restituisce il numero di giorni trascorsi tra le due date. Ad esempio,

```
date_difference(2011, 1, 1, 2012, 1, 1)
```

restituisce 365.

Si scriva quindi il file `test_date.cpp` che definisca la funzione `main` che: (i) legge le date dallo standard input; (ii) scrive sullo standard output il numero di giorni di differenza.

Si tenga conto che:

- il mese è indicato da un numero compreso tra 1 e 12;

- il giorno è indicato da un numero compreso tra 1 e 31 (alcuni mesi hanno un numero inferiore di giorni);
- quando la prima data è successiva alla seconda il risultato dovrà essere negativo;
- alcuni anni sono bisestili (opzionale).