

# Esame di Metodi Computazionali della Fisica, modulo I

17 dicembre 2008, ore 16.00

Tempo previsto per la prova: 2 ore

## Premesse

Il compito è strutturato in due parti, una relativa a *Mathematica* e una al *C++*.

Abbiamo valutato che la risoluzione completa della parte di *Mathematica* dovrebbe richiedere non più di 40 minuti, mentre per la parte di *C++* dovrebbero servire al massimo 1 ora e 20 minuti. Abbiamo distinto domande a cui è obbligatorio rispondere da altre che sono invece facoltative.

Lo svolgimento esatto delle sole **parti obbligatorie** comporta un punteggio massimo di **24**. Ogni punto **facoltativo** svolto comporta l'aggiunta del numero di punti indicato (1 o 2 a seconda della difficoltà), fino ad massimo di **10 punti** aggiuntivi. Il voto sarà dato dalla somma dei punti sopra indicato e la lode sarà assegnata con un punteggio maggiore di 30.

Per lo svolgimento del compito create nel vostro account una directory `/home/username/esame17dic08/` in cui scriverete i vostri programmi.

Una volta concluso l'esercizio, create un unico file `tgz` in questo modo:

```
tar cvz *.m *.hpp *.cpp Makefile > [Matricola]_[data].tgz
```

Conservate comunque una copia del codice per sicurezza.

## Mathematica

Lo svolgimento dei due problemi di *Mathematica* richiede la scrittura di un file `.m` oppure `.nb`. Quando le istruzioni contenute in questo file/notebook vengono eseguite: 1) non devono comparire messaggi di errore e 2) devono venire prodotti i risultati richiesti, la cui correttezza è necessaria per una positiva valutazione dell'esercizio.

## C++

Nell'esercizio di *C++* è prevista l'implementazione di una classe *polinomio* (di primo grado) sul campo dei numeri *razionali*. Sarà dunque necessario implementare per il campo dei razionali i corretti operatori fondamentali. Nell'implementazione della classe dei razionali, un razionale sarà definito come la coppia ordinata di *interi*, numeratore e denominatore.

La prova è suddivisa in due parti: la prima parte incentrata sullo sviluppo della classe dei razionali ed una seconda parte incentrata sullo sviluppo della classe dei polinomi di primo grado. Ogni parte è suddivisa in punti, i quali possono essere o obbligatori o facoltativi. I punti obbligatori debbono necessariamente essere svolti in sequenza per il corretto funzionamento della libreria mentre i punti facoltativi (a meno che non sia espressamente richiesto) possono essere svolti in qualsiasi ordine.

È consigliato svolgere prima la parte obbligatoria e solo successivamente quella facoltativa. Il codice deve essere **correttamente compilabile** e accompagnato dal rispettivo **Makefile** funzionante. Non saranno accettati codici non compilabili (abbiate cura di commentare la parte di codice che non riuscite sviluppare. In caso di indecisione sul voto tale parte sarà comunque parzialmente valutata). Il codice deve rispondere correttamente alle richieste previste nel main. In sede di valutazione il codice sarà comunque testato per verificarne l'affidabilità.

# Mathematica

## 1. Livello fondamentale di un oscillatore armonico

1) Obb. Lo stato fondamentale per un sistema descritto dall'hamiltoniana

$$H = -\frac{1}{2m} \frac{d^2}{dx^2} + \frac{m\omega^2}{2} x^2$$

ha la forma  $\psi(x) = A \exp(-bx^2)$ . Si calcoli il valore di  $A$  tale da normalizzare l'autostato sull'intervallo  $(-\infty, \infty)$ .

2) Obb. Si ricavi l'autovalore  $E_0$  dell'energia corrispondente a  $\psi(x)$ , calcolando il valore di aspettazione di  $H$  su questo stato

$$E_b \equiv \langle H \rangle = \int_{-\infty}^{\infty} dx \psi^*(x) H \psi(x)$$

e imponendo la condizione di minimo rispetto a variazioni del parametro  $b$  (ovvero  $E_0$  si ricava dalla soluzione dell'equazione  $\frac{dE_b}{db} = 0$ ).

## 2. Potenziale generato da un dipolo elettrico

3) Obb. Due cariche elettriche  $q_1$  e  $q_2$  giacciono nel piano  $xy$ , con coordinate  $(x_1, y_1)$  e  $(x_2, y_2)$  rispettivamente.

Si definiscano due funzioni che descrivono, in un punto generico  $(x, y)$ , una il potenziale coulombiano e l'altra il campo elettrico, generati dalle due cariche.

4) Obb. Si definiscano due funzioni che permettano, per coordinate arbitrarie delle cariche, anch'esse arbitrarie,  $q_1$  e  $q_2$ , di disegnare:

- 1) la superficie del potenziale per  $x$  e  $y$  compresi in  $[-5, 5]$ ;
- 2) le curve di livello del potenziale nello stesso intervallo.

5) Obb. Si disegni la superficie del potenziale e le corrispondenti curve di livello con:

- 1)  $q_1 = 1, q_2 = +1, (x_1, y_1) = (-3, -3), (x_2, y_2) = (1, 1)$
- 2)  $q_1 = 1, q_2 = -1, (x_1, y_1) = (-3, -3), (x_2, y_2) = (1, 1)$
- 3)  $q_1 = 1, q_2 = +1, (x_1, y_1) = (-3, -3), (x_2, y_2) = (-2.7, -2.7)$

6) Fac. 3 pt Si definisca una terza funzione che disegni sul piano  $xy$  l'orientazione del campo elettrico punto per punto.

A questo scopo si carichi il package `Graphics`` e si utilizzi la primitiva grafica `Arrow` (l'help appare dopo aver caricato il package). Si generi una matrice `M` 15x15 di frecce, di lunghezza opportunamente ridotta in funzione delle dimensioni del plot, orientate come il campo elettrico in quel punto (*suggerimento: nel punto  $i$ -esimo la freccia va da  $(x_i, y_i)$  al punto  $(x_i, y_i) + \vec{E}(x_i, y_i)$  ed è opportunamente normalizzata*). Si visualizzi il risultato applicando `Show[ Graphics[ M ] ]`. Si sovrappongano, nell'ordine, il plot delle curve di livello e il plot con l'orientamento del campo elettrico.

## C++

### • I PARTE: Classe dei Numeri Razionali

#### (a - Obb) Definizione della classe $Q$

Si costruisca una classe per gestire i numeri razionali (nel seguito indicato con  $Q$ )<sup>1</sup>. Si inserisca nella classe un membro *status*, di tipo *int* e di sola lettura, che possa assumere esclusivamente due valori:

- \* *status* = 0 oggetto valido
- \* *status* = 1 oggetto invalido

tale membro andrà utilizzato per gestire operazioni aritmetiche, tra elementi di tipo  $Q$ , mal definite, ad es  $1/0$  ecc.

#### (b - Obb) Costruttori e distruttore

Definire i seguenti costruttori:

(b.1) costruttore di default;

(b.2) costruttore della forma:

$Q(int\ n, int\ m)$

per costruire un numero razionale  $\frac{n}{m}$ .

(b.2) costruttore della forma:

$Q(int\ n)$

per costruire un numero razionale  $\frac{n}{1}$  (isomorfo al numero intero  $n$ ).

(b.3) costruttore di copia;

(b.4) il distruttore;

*Tip:* scegliere come valori di default 0 ed 1 rispettivamente per il numeratore ed il denominatore. Ricordate di aggiornare lo *status*.

#### (c - Fac, 2p) Forma minima

Implementare una funzione

`void Forma_ridotta(void);`

che consenta di ridurre alla sua forma minima la frazione, ovvero alla sua forma  $\frac{n}{m}$  tale che non esista alcun intero  $s > 1$  per cui  $n = s \cdot \hat{n}$  e  $m = s \cdot \hat{m}$ .

*Tip:* Si consiglia di usare la funzione che calcoli il resto della divisione di due interi positivi  $s$  e  $t$  ( $s/t$ ) % che restituisce l'intero  $q$  tale che:  $s = k \cdot t + q$  con  $0 \leq q < t$  e  $k$  intero  $\geq 0$ . Poiché il calcolo del resto va fatto tra interi positivi, è consigliato l'uso della pseudofunzione

```
#define ABS(X) ((X > 0)?X:-X)
```

```
void Forma_ridotta(void) {
```

```
...
```

```
};
```

```
#undef ABS
```

che restituisce il valore assoluto del parametro (ad esempio: `int a = -3; int b = ABS(a);`).

---

<sup>1</sup>Si ricorda che un numero razionale é definito come una coppia ordinata di generici numeri interi  $n$  ed  $m$

(d) Operatori

All'interno della classe  $Q$  ridefinire gli operatori seguenti:

(d.1 - Obb) gli operatori  $+$ ,  $*$ ,  $-$  e  $/$  tra oggetti di tipo  $Q$ ;

(d.2 - Obb) l'operatore  $=$  tra oggetti di tipo  $Q$ ;

(d.3 - Fac, 0.5p) l'operatore  $==$  tra oggetti di tipo  $Q$ ; si badi al fatto che

$$\frac{n \cdot p}{m \cdot p} = \frac{n}{m}.$$

*Tip:* Ricordate che l'operatore  $==$  non modifica la classe. Per svolgere correttamente questo punto è necessario aver già svolto il punto (c).

(d.4 - Obb) l'operatore  $^$  (elevamento a potenza) tra un oggetto di tipo  $Q$  (left) ed un intero (right);

(d.5 - Fac, 0.5p) gli operatori  $+$ ,  $*$ ,  $-$  e  $/$  tra un oggetto di tipo  $Q$  (left) e un intero (right);

(d.6 - Fac, 0.5p) l'operatore  $=$  tra un oggetto di tipo  $Q$  (left) ed un intero (right);

(d.7 - Fac, 0.5p) l'operatore  $==$  tra un oggetto di tipo  $Q$  (left) ed un intero (right);

(d.8 - Fac, 0.5p) gli operatori  $+$ ,  $*$ ,  $-$ ,  $/$  e tra un intero (left) ed un oggetto di tipo  $Q$  (right);

(d.9 - Fac, 0.5p) l'operatore  $==$  tra un intero (left) ed un oggetto di tipo  $Q$  (right);

• **II PARTE: Classe dei Polinomi**

(e - Obb) Definizione della classe polinomio

Costruire una classe per gestire i polinomi di 1 grado ( $Poli\_1\_Q$ ):  $ax + b$ , con  $a$  e  $b$  oggetti di tipo  $Q$  e  $x$  a valori in  $Q$ .

(f - Obb) Costruttori e distruttore

Definire i seguenti costruttori:

(f.1) un costruttore di default;

(f.2) un costruttore della forma:

$$Poli\_1\_Q(const\ Q\ \mathcal{E}a, const\ Q\ \mathcal{E}b)$$

(f.3) un costruttore di copia;

(f.4) il distruttore;

(g - Obb) Funzione per valutare un polinomio

Definire una funzione per valutare il polinomio di primo grado per un generico valore di  $x$  ( $x$  oggetto di tipo  $Q$ ).

(h - Fac, 2p) Funzione per trovare la radice di un polinomio di primo grado

Definire una funzione per trovare la radice del polinomio nel campo dei numeri razionali, ovvero nel campo degli oggetti di tipo  $Q$ .

La funzione (membro della classe) sia definita nella seguente maniera:

$Q\ Solve(void)$  ;

*Tip:* Si consiglia di implementare l'operatore che restituisce l'opposto di un numero razionale. Tale operatore sarà esterno alla classe e avrà la forma

$Razionale\ operator-(const\ Razionale\ \mathcal{E}q)$ ;

(i - Obb) Programma main

Scrivere un programma main in cui si istanzino due oggetti di tipo  $Q$ , chiamati  $a$  e  $b$ , con

$$\begin{aligned} a &= \frac{3}{4}, \\ b &= \frac{1}{2}, \end{aligned} \tag{1}$$

Si istanzi un oggetto di tipo  $Poli_1_Q$  e si valuti il polinomio nel punto  $x = \frac{2}{3}$ . Nel caso in cui sia stato svolto il punto (h), trovare la radice in  $Q$ . I risultati vanno stampati a video, e nel caso sia stata svolto il punto (c), stamparne anche la forma ridotta.

□