

Esame di Lab. Informatica II anno - A.A. 2007 - 2008

Stefano.Lacaprara@pd.infn.it

- Di seguito trovate N esercizi simili a quelli che abbiamo svolto a lezione.
- Voi ne risolvete **due** a vostra scelta e lo portate all'esame con penna-USB/CD/etc...
- Inoltre dovete sapere descrivere *qualitativamente* come risolvereste altri 2 esercizi a vostra scelta.
- All'orale mi mostrate cosa avete fatto e perché.
- Se l'esercizio prevede l'implementazione di una classe, dovete anche scrivere un piccolo esempio (*main()*) di uso della classe.
- Se avete problemi di comprensione del testo, mettetevi in contatto con me.
Stefano.Lacaprara@pd.infn.it

Esercizi

1. Scrivere una classe che rappresenta una *particella* elementare generica. Le caratteristiche sono: massa, nome, velocità, energia, se decade o meno. Poi una seconda classe che rappresenta un *elettrone* (massa= .511 MeV, non decade) e infine una che rappresenta un *muone* (è una specie di elettrone pesante): massa= 106.7 MeV, decade e ha una vita media $\tau = 2.2 \mu s$.
2. Scrivere una classe che rappresenta i numeri razionali (rapporto tra due interi) e implementa le operazioni elementari (+,-,/,*, stampa, inverso, etc...).
3. Scrivere una classe che rappresenta matrici 2x2, con operazioni elementari (+, prodotto per scalare, determinante, trasposta, etc...)
4. Scrivere una classe che rappresenta una matrice 2x2 ad elementi complessi (usare i complessi fatti a lezione) con operazioni elementari come nell'esercizio precedente.
5. Disegnare funzione così definita:

$$f(x) = \begin{cases} 0 & x \leq 0 \\ e^{-x} \cdot \sin^2(x^2) & 0 < x < 10 \\ 0 & x \geq 10 \end{cases}$$

e calcolarne l'area con il metodo MonteCarlo oppure con il metodo dei trapezi (definizione di integrale).

6. Creare e visualizzare un istogramma i cui (30) bin hanno i seguenti valori:

```
float bin[30] = {144, 391, 607, 801, 1036, 1111, 1381, 1495, 1670, 2092,
3344, 4879, 4988, 3524, 2372, 2126, 2018, 1897, 1979, 1755,
1721, 1617, 1541, 1398, 1208, 1047, 752, 597, 378, 131};
```

(trovate un file con questi valori all'indirizzo:

<http://www.pd.infn.it/~lacaprar/Didattica/C++/DatiEx6.data>)

Fare un fit gaussiano al picco presente usando una opportuna funzione (polinomio di grado N) per modellare il fondo.

7. Creare un istogramma mono-dimensionale e riempirlo con N numeri casuali distribuiti secondo una distribuzione gaussiana (o altra a vostra scelta). Visualizzarlo e fare un fit. Creare poi un secondo istogramma e riempirlo con la distribuzione integrale del primo,

$$F(k) = \int_k^{+\infty} f(x)dx$$

(usare il metodo `double TH1::Integral(int binx1, int binx2) const`). Visualizzare anche il secondo istogramma.

8. Generate una distribuzione gaussiana utilizzando un generatore di numeri uniforme e il teorema del limite centrale. Riempire un istogramma, visualizzarlo e fare un fit.
9. Stimare l'area di cerchio di raggio unitario (π) con il metodo MonteCarlo. Ripetere la stima M volte e riempire un istogramma con i risultati. Stimare l'errore associato a π .
10. Si ripeta l'esercizio precedente con diversi valori per il numero di sassi utilizzati (N) e si faccia un grafico del valore dell'errore su π in funzione di N . Fare infine fit su tale grafico con una funzione da scegliere in base all'andamento che vi aspettate di σ_π vs N .
11. Il frattale di Julia è definito in modo simile a quello di Mandelbrot, ma con un ciclo diverso che dipende da un parametro esterno c .

- Si supponga considerare una piccola porzione attorno all'origine di un piano complesso di intervallo $(-2, -2) \div (2, 2)$.
- Si scelga un numero complesso c .
- Si sostituiscano, per ogni punto considerato, le corrispondenti coordinate complesse alla variabile z .
- Si calcoli il valore di $z = z * z + c$.
- Se si trova che la distanza di z dall'origine è maggiore di 10, si salti al passo 7.
- Se no, si incrementi di 1 un contatore e si torni al passo 4 se il contatore ha un valore inferiore al numero di iterazioni massime prefissato (per esempio 100).
- Si colori il punto di un colore diverso a seconda del valore del contatore.
- Si azzeri il contatore e si ritorni al passo 3 per calcolare il colore del prossimo punto.
- Il procedimento avrà termine quando tutti i punti interessati saranno stati processati nel suddetto modo.

Disegnare con root gli insiemi di Julia per

- $c = (0, 0.65)$
- $c = (0, 0.60)$

e stimarne le aree (facoltativo).

12. Riempire un istogramma con numeri casuali distribuiti secondo la funzione

$$x^2 \cdot \sin(x) + \text{gaus}(0, 1)$$

, nell'intervallo $[-4, 4]$, dove $\text{gaus}(0, 1)$ è la funzione gaussiana con media 0. e $\sigma = 1.$, usando la tecnica dello scarto (o rejezione).

13. Usando il container `map` delle STL, scrivere una classe che rappresenta un semplice elenco telefonico (associazione nome, numero di telefono), implementando un insieme ragionevole di metodi, quali inserimento, ricerca, cancellazione, etc ...
14. Scrivere una simulazione di un sistema dinamico costituito da tre pianeti (es Sole, Terra, Luna) sottoposti alla reciproca attrazione gravitazionale, in uno spazio bi-dimensionale (facoltativo tri-dimensionale). Calcolare l'energia potenziale, cinetica e totale del sistema, e farne un istogramma in funzione del tempo. Idem per momento angolare totale (facoltativo).
15. Una lista a puntatori è un modo per avere una collezione di oggetti di cardinalità arbitraria, allocati solo al momento di reale necessità (senza, tuttavia, la proprietà di contiguità in memoria). Essenzialmente gli elementi della lista sono una coppia `{oggetto, indirizzo}`. L'`oggetto` rappresenta il dato che interessa memorizzare, mentre l'`indirizzo` memorizza l'indirizzo della coppia `{oggetto, indirizzo}` successiva.

Al momento di creare un nuovo elemento della lista, si registra il suo indirizzo nella variabile `indirizzo` dell'ultimo elemento della lista. Per poter scorrere la lista, quindi, è sufficiente tenere da qualche parte l'indirizzo della prima coppia, per sapere da dove partire. Per comodità, si può anche tenere da parte l'indirizzo dell'ultima coppia, così che per aggiungere un nuovo elemento non sia necessario scorrere tutta la lista alla ricerca della coppia che ha la variabile `indirizzo` nulla.

Si richiede di:

- scrivere una classe per una lista a puntatori contenente `float`. La classe della lista deve avere un metodo `void push(float)` per aggiungere un nuovo oggetto alla fine lista, e un metodo `float pop()` che restituisca il primo elemento della lista e lo *rimuova* dalla lista stessa.

In questo modo il container è di tipo *FIFO* (**F**irst **I**n **F**irst **O**ut): il primo elemento inserito con `void push(float)` è il primo ad essere restituito con `float pop()`.

- risolvere il punto precedente, (facoltativo) lasciando *template* il tipo di dato da gestire (quel che prima era `float`).

16. svolgere l'esercizio precedente, facendo in modo che il metodo `float pop()` restituisca e rimuova l'ultimo elemento della lista.

In questo modo il container è di tipo *LIFO* (**L**ast **I**n **F**irst **O**ut): l'ultimo elemento aggiunto con `void push(float)` è il primo ad essere restituito con `float pop()`.

17. Si consideri una casa automobilistica che produce un certo numero di modelli diversi. Si vuole avere una gerarchia di classi che rappresentano questi modelli, sfruttando in modo opportuno un approccio Object Oriented:

- implementare classi per almeno due modelli differenti di automobile;
- è sufficiente scegliere poche proprietà delle automobili per il database, ad esempio il colore della carrozzeria, il numero di posti; un optional comune a tutti i modelli ed un optional disponibile per alcuni modelli solamente.
- implementare uno schema con un livello di ereditarietà;

Un concessionario compra un certo numero di auto di diversi modelli, e ha il problema di avere una lista sempre aggiornata di tutte le auto che possiede. Scrivere una classe che è in grado di contenere un numero generico di automobili dei diversi modelli e in grado di stampare la lista completa e il numero di auto di un dato colore. (Usare le proprietà polimorfiche delle classi derivate).