COMPUTING METHODS FOR PHYSICS
28 JANUARY 2019

You must submit your exam by **Wednesday Jan 30 at 16:00** following the instruction at http://www.roma1.infn.it/people/rahatlou/cmp/

Composite Pattern for particles

Use the composite pattern to implement a polymorphic hierarchy of two classes **`Particle`** and **`CompositeParticle`**. (15 points)

- Each particle must have 2 data members **`mass`** and **`momentum`** (use TLorentzVector class from ROOT)
- Implement add and remove methods for **`CompositeParticle`**
- Implement the momentum and energy methods for **`Particle`** and **`CompositeParticle`**
- Overload the **+** and **<<** operators for **`Particle`** and **`CompositeParticle`**
- Write an application **`app.cc`** to test your classes. For example define to particles **`pion`** and **`kaon`** and add them to create a composite particle **`D0`**.
- No additional functions, classes or applications are needed. If additional functions are present, they will be evaluated and possible mistakes will decrease the total score.

Provide instructions for compiling your code in the comments at the beginning of **`app.cc`**.

Evaluation will be based on: successful compilation, correct use of C++ syntax, polymorphism, return type and arguments of functions, data members and interface of classes, misuse of void functions, use of unnecessary C features, correct kinematics.

Extra (2 points): implement the **`boost`** method for **`Particle`** and **`CompositeParticle`** with adequate return type and argument(s) by delegating the calculation to TLorentzVector. NB: Do not implement a special case of boost.

NB: You cannot replace a mandatory part with the extra question. Extra will be considered only if all mandatory parts completed.

Computing Methods for Physics
28 January 2019

Random Walk in 3D (15 points)

Implement a random walk in 3 dimensions in python using the NumPy and SciPy libraries and matplotlib for plotting. You can write a standalone python program or a jupyter notebook.

The initial position of the particle is at the origin (0,0,0). Simulate 100 random walks each of 1000 steps. At each step the particle can only move by 1 (arbitrary unit) along one of the x,y,z axis (positive or negative directions) with equal probability of 1/6 (it cannot move along the diagonal).

1. Plot the distribution of the maximum distance reached in each experiment
2. Provide a dictionary with the final 3D position of the particle for each experiment when the final distance of the particle is less than 10 (arbitrary units) from the origin.
3. Print the fraction of experiments with final distance < 1
4. Visualise the trajectory of the particle for the experiments with the final position closest to the origin.

Extra (2 points): use animation to visualise the trajectory of the particles only for the experiments where final distance < 10.

Suggestions: You can use a tuple to represent the 3D coordinates.

Evaluation will be based on: Use of python features and data structures, comprehensions, NumPy objects, instead of a simple translation to python of a C program, labels and clarity of plots.

NB: You cannot replace a mandatory part with the extra question. Extra will be considered only if all mandatory parts completed.