COMPUTING METHODS FOR PHYSICS
6 JULY 2020

You must submit your exam by following the instructions at http://www.roma1.infn.it/people/rahatlou/cmp/

Part 1: Bragg Peak (C++)

Provide a class **`Particle`** characterised by

- mass (in GeV)
- charge (in units of e)
- Three momentum

with proper data members, constructor, and accessor functions. Implement the following 3 member functions (with proper arguments if needed and proper return type): **`beta()`**, **`gamma()`**, **`betagamma()`**.

Provide a class **`Material`** characterised by

- Density δ (in g/cm^3)
- Atomic mass A
- charge Z (in units of e)
- mean ionisation energy <I> (in eV)

with proper data members, constructor, and accessor functions. Implement the member function **`dEdx()`**, using a **`Particle`** object as argument, to compute the mean energy loss by ionisation with the Bethe-Bloch formula.

Provide a **`bragg.cc`** application using these classes, to compute the mean energy loss for a proton, an α particle, and an electron of momentum p = 7 MeV, as a function of the penetration depth x (in cm) in water.

- Print the position of the Bragg peak for each particle on the screen

As a reminder, the position of the Bragg peak is the depth x a particle has the maximum energy loss per unit length dE/dx.

Bonus: provide a member function **`BraggPeak()`** (with proper return type) for the class Material that takes a **`Particle`** as input and returns the position of the Bragg Peak, using properly the other class methods.

Evaluation will be based on: correct C++ syntax, proper return type and arguments of functions, data members and interface of classes, unnecessary void functions, correct mathematical operations, correct physics calculation and units.

Useful data:

	density [g/cm^3]	<I> [eV]	E_c [MeV]	X_0 [cm]	δ
H_2O	1.0	80	78	36	9

Part 2: Plots (ROOT or Python)

Use ROOT or python to plot the average energy loss dE/dx as a function of the penetration depth x for the three particles in water.

- choose the proper range to show the plot for all three particle
- overlay the distributions for all particles in one plot
- use different colours or/and line types to differentiate the 3 particles
- add legend for each particle and labels with units

If you use python, you must store the output of the C++ program and read the values in the python program.

Evaluation will be based on use of python features and data structures, python or ROOT classes, labels, units, clarity ad correctness of plots.