LHC Programming Prep Assignment 3 : Memory, Loops, and Classes

Sam Meehan

Due : 19 February 2017 (I won't be back until then to review it)

Notes

1. You should submit your work in the form of either a text file or compileable source code
2. All submissions should have the format : `FIRSTNAME_LASTNAME_Date_Description.txt/.cpp/.h`
3. There is no need to submit screen shots of the result of your program running. I will compile and run your code if necessary.
4. If you submit code, ensure that it compiles. Code that is not yours and doesn't compile is extremely frustrating to work with.

Question 1 Here is some practice storing vectors and working with memory addresses and implementing loops. You must write a piece of code that creates a vector of floats and fills it within a loop. During each iteration of the loop, the user will have to enter commands via the `std::cin` cal that direct the loop to either terminate, and break out of the loop, or proceed to adding another entry onto the vector. If the command is to fill another entry, then the entry should be a random number (you will have to self-educate as to which library in c++ allows you to generate random numbers) that is evenly distributed between 0 and 1. After filling this entry onto the vector, print out the entire contents of the vector, including the entry at each spot and its location in memory (You will probably need a loop within the loop for this). After you have programmed, compiled, and executed this, describe how exactly c++ is allocating memory when you create and expand the vector you have created.

Question 2 In this question you will practice the creation of classes. In HEP, one of the most common types of things we work with are four vectors when describing the kinematics of particles at the LHC and have the form $(E, p_x, p_y), p_z)$. In this exercise, write a piece of code in which you do the following.

1. Define a class that represents a four-vector. Call this class $FourVector$
2. Constructor : Should be implemented to force the user to initialize the members of the class
3. Class Members : This should contain fundamental memory types that represent the energy and (three) momentum of a given particle. Do **not** use a vector to represent energy or momentum. Call these members E, px, py, and pz.
4. Class Method 1 : Should be called and replace the contents of the full four vector with the E, p_x, p_y, and p_z that you specify as arguments. Call this function $SetFourVector$
5. Class Method 2 : Should be called and return the invariant mass of the four vector (as a float). Call this function $Mass$
6. Class Method 3 : Should be called and return the *pseudorapity* (η) of the four vector (as a float). Call this function Eta
7. Class Method 4 : Should be called with one argument that is *another four vector* and calculate the difference in the azimuthal angle ($\Delta\phi$) between the two three-momenta of these vectors. Call this function $DeltaPhi$.
8. Class Method 5 : Should be called with one argument that is *another four vector* and calculates the difference in the *pseudorapity* (η) of the two four-vectors. Call this function $DeltaEta$.

You should implement all of this within a single `main()` function such that I can compile it via `g++` as we have been using and seeing these past sessions. You can organize your code as you like, perhaps using a header to define the struct and make the function prototypes, but the code must be well commented, use appropriate indentation to make it readable, and be readable.

Question 3
From Project Euler, describe the procedure or algorithm that you would use to solve question 8.

"The four adjacent digits in the 1000-digit number that have the greatest product are 9 × 9 × 8 × 9 = 5832.

73167176531330624919225119674426574742355349194934
96983520312774506326239578318016984801869478851843
85861560789112949495459501737958331952853208805511
12540698747158523863050715693290963295227443043557
66896648950445244523161731856403098711121722383113
62229893423380308135336276614282806444486645238749
30358907296290491560440772390713810515859307960866
70172427121883998797908792274921901699720888093776
65727333001053367881220235421809751254540594752243
52584907711670556013604839586446706324415722155397
53697817977846174064955149290862569321978468622482
83972241375657056057490261407972968652414535100474
82166370484403199890008895243450658541227588666881
16427171479924442928230863465674813919123162824586
17866458359124566529476545682848912883142607690042
24219022671055626321111109370544217506941658960408
07198403850962455444362981230987879927244284909188
84580156166097919133875499200524063689912560717606
05886116467109405077541002256983155200055935729725
71636269561882670428252483600823257530420752963450

Find the thirteen adjacent digits in the 1000-digit number that have the greatest product. What is the value of this product?"