C++ Workbook

Ettore Speziale

In the following sections you can found some assignments, each related to a particular aspect of the C++ language. The assignments are more or less of the same difficulty of the ones you are required to solve during the exam.

1 Object Oriented

The Scrooge McDuck corporation needs to continuously observe the stock market in order to compete with other rival corporations. Data for stock market analysis is provided by the Huey, Dewey, and Louie Inc., that asks 1000$ for each access to data.

Due to is greed, Scrooge McDuck does not want to pay 1000$ too many times, so he assign you the design of a new stock market analysis tool:

- when a new analysis is needed, a command containing analysis code is sent to the analysis server
- the analysis server buffers commands, until a fixed threshold is reached
- at that point, a command representing all pending analysis is built, and sent to Huey, Dewey, and Louie Inc., thus allowing to pay its fee only once
- a special command forcing the execution of all pending analysis can be sent to the analysis server

You are required to sketch the solution of the problem using C++.

2 Operator Overloading

The Orion Crew Exploration Vehicle is a new manned aircraft being built by NASA to replace the Shuttle Space Transportation System. It is equipped with state of the art computers in order to increase the aircraft reliability.

The implementation of the flight-control systems has been commissioned to your company. You are in charge of providing a software-based fault-tolerant implementation of single precision floating point arithmetic to be used by software running on flight-control CPUs.

Your boss asks you to sketch an initial version of the library with the following features:

- the base data type is a floating point number, whose value is replicated three times

$$fp = \langle fp_1, fp_2, fp_3 \rangle$$

- fp is valid if at least two of its components are equal
- performing an operation $fp_a \oplus fp_b$ requires first performing the operation $\oplus$ on corresponding replicas

$$\forall_{i \in [1,2,3]} \; fp_a(i) \oplus fp_b(i)$$

  and then selecting the overall result using a voting-based policy
- supported operations are sum, difference, multiplication, and division
- if during an operation a number becomes invalid, execution must be stopped

You are required to design a prototype of the library using the C++ language.

3 Templates

The Solaria corporation is specialized on producing advanced robotic systems. They are assembled in highly automated factories consisting of a sequence of stations, interleaved by buffers.

Currently, the system controlling the assembly line represents buffers using an array of fixed size. After a little chat with Mr. Karp, your boss decided to investigate whether using a circular buffer can improve the performance of the assembly line.

A circular buffer is linear container of fixed size, indexed by two pointers: the *head* points to the first used element of the container, while the *tail* points to the first available element.

The container supports the following operations:

- *push* adds an element to container tail. If the container is full, insertion is not performed
- *pop* removes an element from container head. If the container is empty, remove is not performed
- *front* returns the first element of the container
- *back* returns the last element of the container
- *capacity* computes the number of elements the container can contains
- *size* calculates the number of elements currently stored in the container
- *empty* checks whether the container is empty

You are required to implement the circular buffer using C++.

4 Traits

Los Tres Amigos Inc. main product is a renderer for Java classes. Given an instance of java.lang.Class it print to standard output its name, followed by the list of attributes and methods. Attributes and methods are preceded by a character identifying their scope: + for public visibility, # for protected visibility and - for private visibility.

The company strategy is to develop a version of the renderer for C++. With respect to the Java version, the renderer should not use reflection to obtain information about a given instance. Moreover, one the major goal of the tool is to provide an interface to render anything that looks like a class. For instance, it would be possible rendering a **struct** or an **enum**.

You are required to implement a prototype of the tool.

5 Allocators

After years of developments, the code-base of GCC & Sons is full of memory leaks. In order to deal with new competitors, like LLVM & Clang, the board of directors decide for an extensive review of the code-base.

The first analyzed product is the editor Scame. The main source of leaks is the representation of characters. Indeed, Scame is a WYSIWYG editor and each character is represented using its own instance encoding both the character itself and the formatting options.

Your boss order you to reduce the amount of memory consumed by Scame by applying the flyweight design pattern to character representation.

Your first step is evaluating the approach by designing a C++ library that allows to format a string an render it to standard output. The character set includes alphanumeric characters plus the space character. Each character is internally represented in the lowercase form, while it can be formatted in lower or upper case. Obviously, the way digits and space are formatted is unique.

6 Parallelism

Mandrake likes betting on horse races at San Siro. In order to increase the betting efficiency he decide to code an horse races simulator.

Each horse i is characterized by a base speed v_i and a competitive factor $c_i : 0 \leq c_i \leq 1$. All horses runs on private lanes with the same length. At each time step t, the i-th horse advance along its lane to the position $x_i(t)$, taking into consideration the behaviours of all the other horses:

$$x_i(t) = \begin{cases} x_i(t-1) + v_i c_i \sum_{j \neq i} \frac{1}{d(i,j)} & \text{for} \quad t > 0 \\ 0 & \text{for} \quad t = 0 \\ \bot & \text{for} \quad t < 0 \end{cases}$$

Where $d(i,j)$ is the distance between i-th and j-th horse on the lane at the previous time step – the advantage of the i-th horse over the j-th horse. The simulation tool simply updates the position of each horse on the lanes, until all of them reach the end of the race.

Mandrake coded a serial version of the simulator, but now it wants to parallelize it in order to speedup simulation time. You are required to code a prototype of the simulator in C++.