Project One

Programming Fundamentals

***Objectives of this lab:***

* Learn some basics about computer systems
* Learn about problem solving and programming
* Illustration of the problem solving and programming - A simple C++ program

**Program Design Process**

Problem solving is usually broken into two major phases:   
1) Problem solving phase, and 2) Implementation phase.

Phase (1) - In the first phase, you will take three steps:   
**Step I:** You will define the problem that you want to solve, clearly.   
**Step II:** You will design an algorithm that is precise and very well thought to solve the problem, and   
**Step III:** You will test your algorithm on paper.  Your algorithm should work correctly, before you can write a program for it.

Phase (2) -In this phase, you will take two steps:   
**Step I:** Translate your algorithm to C++ language.  If you have a correct and precise algorithm, the translation should be almost line-by-line.  This translation must be correct and free of:

* A) Syntax errors, which are the errors resulted from incorrect use of the programming language syntax or violation of syntax rules.
* B) Computations that are not possible, such as dividing by 0, and
* C) Errors made by the programmer.  Such errors are those made by using wrong signs or arithmetic operators.

**Step II:** Test the program to make sure it produces the correct results.  Make sure your test cases are different.  The only way to correctly test a program is to have many different test cases.

**Part 1**: **Algorithms and flowcharts**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Exercise 1.1**

In the following diagram write a question in the blank box that best fits the statements that have appeared after the box. Assume that just before the blank box you have read the values for *x* and *y*.

Is the value of y = 0 ?

Division by zero is not defined.

The answer is x/y

yes

no

In the following diagram write an English statement in the blank box that best fits the statement that has appeared before the box.

Is y a negative number?

The root of negative doesn’t exit

The answer is √y

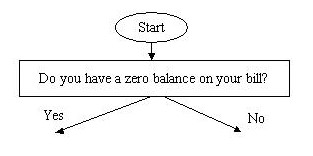
yes

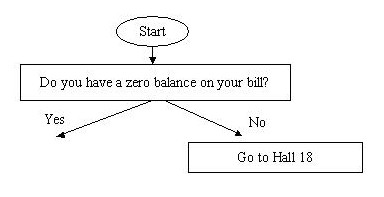
no

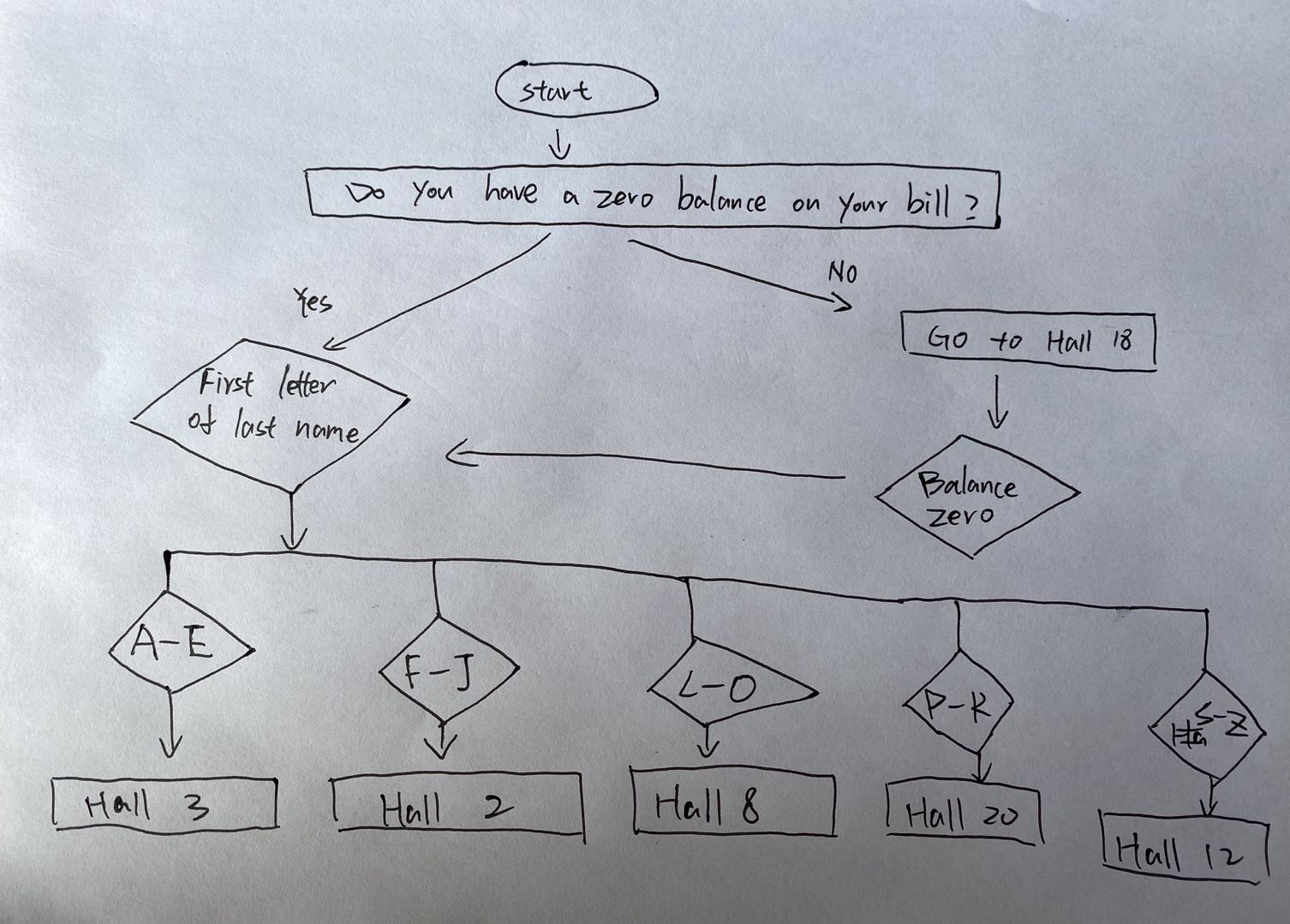
**Exercise 1.2**   
Suppose you are helping the university registrar office with the registration process. You are to send students to six different halls depending on the first letter of their last names and the balance that has appeared on their bills.  Here are the criteria you will use to separate them:   
Students with balance zero, Letters:   
A-E in Hall 3, F-J in Hall 2, L-O in Hall 8, P-R in Hall 10, and S-Z in Hall 12.

Students with a non-zero balance go to Hall 18. These students can go back to register once they have a zero balance on their bills.

Draw a flowchart to show the steps that will take to solve this problem.







**Part Two: A Simple C++ Program**

A grocery store sells many cases of soft drink every day.  In each case, there are 12 bottles and the store profits 20 cents per bottle. We want to compute the profit that the store has every day of selling soft drink.  We also want to know the profit for selling soft drink in a year. Assume a year is 365 days.

1. **Problem Definition:**   
   Compute the profit that a store has in one day for selling soft drink?   
   Compute the profit that a store has in one year for selling soft drink?
2. dailyProfit = bottles \* 0.2 cents

yearlyProfit = dailyProfit \* 365 days

1. **Program Design - Algorithm**   
   Before we attempt to write the program, let's develop an algorithm for solving the problem.

**Exercise 2.1**  
Design the **algorithm** for this problem written in Pseudo Code (plain English).  Remember your algorithm must be precise. This algorithm must be translated to C++ to obtain the program.

The daily profit will be the number of cases sold which is 12 bottles per case times 20 cents.

The yearly Profit will equal the daily profit times 365 days.

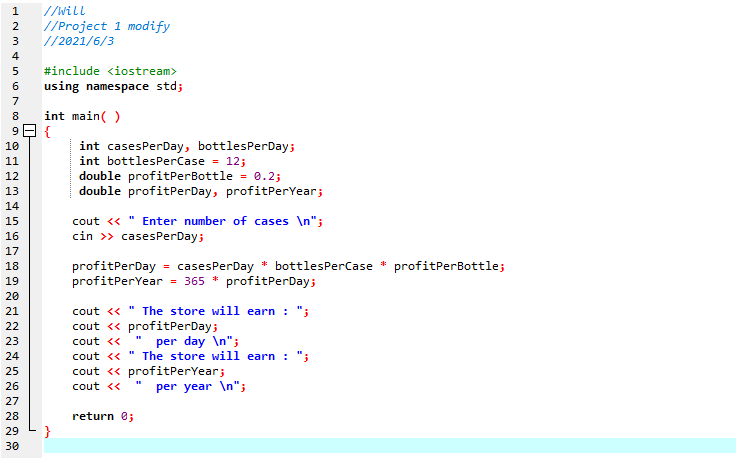
1. **Desktop Testing**- **Trace Table**  
   Now that you have the algorithm, create a **Trace Table** to test your algorithm and see if it works on paper. Remember, your table should have two columns, one for the variables and the other for expected output.

**4) Implementing the Algorithm in C++**

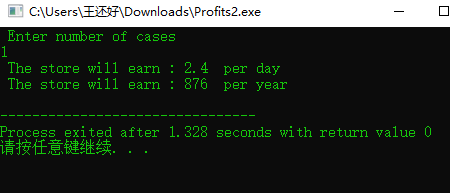
The last thing you need to do is to test the program to make sure it produces the correct results. Assuming the store sells 10 cases per day, what would be the profit per day and per year? You can confirm the answer by hand.

**Exercise 2.2**  
This is where you will translate the algorithm to C++.  Write a program called Profits.cpp that is designed based on the algorithm that is given in the previous part.

// This C++ Program will compute the profit of selling soft drinks

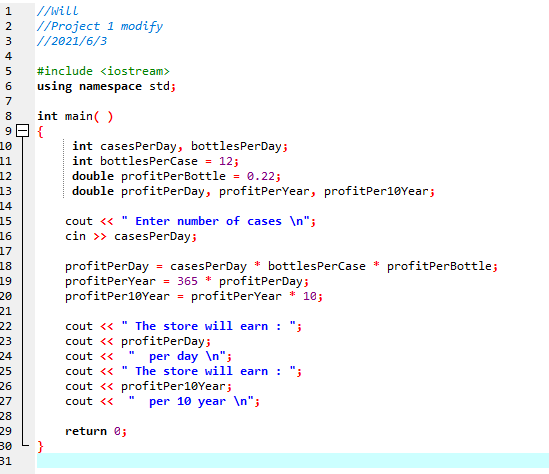


Test Case

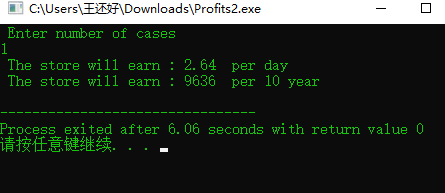


**4) Test the Program**   
The last thing you need to do is to test the program to make sure it produces the correct results. Assuming the store sells 10 cases per day, what would be the profit per day and per year?  You can confirm the answer by hand.

**Exercise 2.3**  
Modify your program so that it uses 22 Cents profit per bottle in the calculations.  This time in your output, display both the number of bottles sold and the profit for one day, one year, and 10 years.  Call your new program Profits2.cpp



Test case



Headers/Task Division Table

All tasks are completed by Will.