Exam-type problems

Note: these problems are similar in length and difficulty to the "big questions" on the exam. The exam typically involves several much smaller questions and several big questions. The smaller questions are similar to the other drill problems and the practice problems.

How to use these problems to study

In order to make the most of these problems, I suggest you work through them as a last step in your studying. By the time you get here, you should be well versed in the syntax of coding and you should be reasonably comfortable writing small questions.

You should treat these questions as though you were writing them during the exam. Keep only your cheat sheet handy, and write out your solution by hand first. Once you're done, put the code into the computer and test it to see if you were correct. Note any issues. If there were logical errors, go back to the relevant sections and practice them. If there were major errors, work through the problem again carefully, noting where you misread the problem, and make sure that you fully understand what went wrong.

How to work with the files: most of the C++ programs will involve file IO. You will notice that I didn't provide these files. On an exam, we typically describe the format of the file and provide some sample lines, but do not provide the whole file. You need to become comfortable working with a description of the file without having access to it. If you want to test, you should make up your own file for testing purposes.

C++

Question 1

Fiona is starting a business selling Lego Robot components. She wants you to make her an inventory database and sales management system. She intends to store product information in a file, "inventory.txt", so that she can turn off her computers at night to save electricity. The first three lines of the file are shown below:

```
3987.45
NXT_BRICK 300.45 200.5 75
COLOUR_SENSOR 57.86 30.8 40
```

On the first line is the amount of cash that Fiona has on hand at the start of the day. After the first line comes the product information. Each product is described on a single line. The first entry is the name of the product. The second entry is the price that she sells the product for, the third entry is the price that she buys the product for, and the final entry is the number of each product that she has in stock. Since she is just starting out, she has decided that she will only sell at most 10 products.

Write a C++ program that performs the following tasks:

1. Opens the file and verifies that it has opened.
2. Reads the file's content into appropriate variables (I suggest using arrays...)
3. Closes the file.
4. In a loop, prompts the user to enter a product by name. If the product is available, they will be sold the product. Update Fiona's cash on hand and inventory. If the product is not available, inform the user.
5. The loop should be exited when the user enters the string "STORE_CLOSED".
6. Fiona wants to ensure that she has at least 10 items of each type on hand at any given time. If any product has fewer than 10 items in inventory at the end of the day and she has enough cash to order more, you should buy enough to ensure that there are 10 items in the inventory. You may decide what to do if she doesn't have enough cash to order all of the items but she can order some of them.
7. Once the loop has been exited, the file should be **re-opened** as an output file, and all of the data should be written back to it.

Question 2

Mike is experimenting with automated marking of assignments. He's choosing to focus on multiple choice problems only. He wrote a file, "answers.txt", which contains the correct answers, one per line. The first three lines of the file are shown below:

```
A
C
D
```

The correct response for question 1 is 'A', for question 2 is 'C', and for question 3 is 'D'. There are at most 30 questions in the assignment. A similar file, "students.txt", contains student answers. The first line in the file is an integer indicating how many students did the assignment. After that, each student's responses are recorded. Students may answer with a single character from A to E, or may enter the character Z to indicate that they chose not to answer the question. The first few lines of the file are shown below:

```
17
A
D
Z
```

Students get 5 marks for each correct answer, -2 marks for each incorrect answer, and 0 marks for any questions they chose not to answer.

1. Write a function that receives a student's answer to a question and the correct answer. Return the score they received on that question.
2. Write a function that receives a total grade that the student received, and any other information you think they need, and returns the student's grade out of 100.
3. Write a main program that performs the following tasks:

- Opens both input files and verifies that they are opened.
- Reads "answers.txt" into an array, and determines how many questions were asked on the assignment.
- Reads the file "students.txt" and outputs the score out of 100 that each student received on the assignment. **Note**: you cannot assume that there are exactly 20 questions on the assignment. You need to figure that out!
- Outputs the class average at the end.

Your main function must call functions 1. and 2. above.

Robot C

Question 1

Captain Star is making a space-age vending machine robot that senses when people are near, then tries to sell them things. A function, giveCandy(), has already been written for you. It gives the person their selected candy. Help Captain Star by writing the following program:

- The vending machine displays a friendly message until someone comes within 100cm of its sonar sensor.
- When someone is in range, the display changes and prompts them to make a selection.
    - If button 1 is pushed, they are given a small candy
    - If button 2 is pushed, they are given a large candy
    - If button 3 is pushed, they are asked if they want a chocolate or a Curly Wurly, and respond using the buttons.
- Once a selection has been made, you must call the function giveCandy.
- You must keep track of how much money has been made. Small candies cost $1, large candies cost $2, chocolates cost $3, and Curly Wurlies cost $4.
- If the touch sensor is pressed at **any** time during the process, the robot must output how much money it has made for 3 seconds, then the program ends.

Question 2

A robot is placed in the bottom left corner of a square room. It is in the standard lab configuration, except the color sensor has been rotated 90 degrees so that, if there is an obstacle directly in front of the robot, the robot can sense its color. In addition, the robot has a Swiffer Sweeper™ cloth attached underneath it. The swiffer cloth is 15cm wide.

The walls of the room are yellow, and obstacles can be either blue or red. A red obstacle is small, and the robot can simply push it over by increasing its motor power to 100% for 1s. During this pushing phase, the robot will not move forward, and it may resume its previous course. A blue obstacle is too big. Once found, the robot must wait for help before continuing. The human supervisor will periodically check on the robot. If it is waiting for help, the human will remove the blue obstacle and push the orange button.

A function, turn90(int dir), has been provided to you. It turns the robot
90 degrees. If dir == 1, it will turn the robot clockwise. Otherwise, it
will turn it counter-clockwise.

Your task is to write two functions and one main program that will enable the robot to sweep the entire floor.

1. State at least one assumption that you will make to solve this problem.
2. State at least one problem with the above plan that will make it difficult for the robot to complete its task.
3. Write the program according to the rules below:

- Write a function that takes in a desired motor power and drives the robot forward at the given power until it hits an obstacle. The function must return the color of the obstacle.
- Write a second, nontrivial function of your choosing that will assist in the task.
- Write a task main that calls your two functions above, and includes any other code that you need to ensure that the robot sweeps the entire floor.