Problem 1.

Incoming transfers at Zoom University are attending new student orientation, where they have just been assigned a nine-digit student ID. After lunch, they must attend three different workshops, each sorted into its own breakout room.

As the head New Student Advisor (NSA), your job is to allocate students into these breakout rooms based on the last digit of their student ID. Their "homeroom" is based on the following guidelines:

- All students with ID's ending in 0, 1, or 3 must report to Breakout Room 1
- All students with ID's ending in 2, 4, or 5 must report to Breakout Room 2
- All students with ID's ending in 6, 7, or 8 must report to Breakout Room 3
- Everyone else will be on break and will go to Breakout Room 4

Write *two* distinct programs that will help you with your job. Do not simply change the name of an identifier, add different annotations, or write the arithmetic in a slightly different way. Your program must contain two different techniques described in lecture. The outputs of your program should look like the following. (Note than the text in parentheses are sample inputs.)

```
Enter your student ID: (104349103)
You should go to Breakout Room (1).
```

This page is intentionally left blank.

This page is intentionally left blank.

Problem 2.

If the following program compiles, write all the possible different outputs that it would generate. If it does not compile, locate the compilation error(s), fix them, and write down the possible outputs that would result from the changes you made.

```
#include <iostream>
using namespace std

int main() {
  string response;
  cout << "Do you live in a heavy-populated area? ";
  cin >> response;

  if (response = "No" || "no");
    cout << "Stay safe" << endl;

  else
  {
    cout << "You are ";
    for (int i = 0; i < 6; i++) {
      switch (i) {
      case 5:
        cout << "ED!" << endl;
        break;
      case 2:
        cout << "R";
      case 1:
        cout << "A";
        break;
      case 4:
        cout << "T";
      case 70:
        cout << "I";
      case 3:
        cout << "N";
        break;
      case 0:
        cout << "Q";
      default:
        cout << "U";
}}}}
```

This page is intentionally left blank.

Problem 3.

You and your friends from U$C and Berkeley decide to have dinner at a fine-dining restaurant in Downtown LA. Unfortunately, they charge you based on the subtotal of your order, a mandatory percent-based fee that varies depending on the school you attend, and an optional tip. UCLA students are charged an additional 3% of their subtotal, Berkeley students are charged 5%, and U$C students are charged 12.4%. Students from other schools are also charged 12.4%. Note that these percentages are based on whatever the *individual* orders. On a positive note, the restaurant offers a $200 discount for each birthday, given two conditions:

- There can be a maximum of two birthdays
  - If there are more than two birthdays, the group will only get a $400 discount
- The lowest they can charge you is $0 (meaning they cannot owe you money)

Also, the city of Los Angeles decided that this restaurant is exempt from the usual 9.5% tax. Write a program that inputs the conditions described above and calculates the *total* amount you and your friends owe the restaurant, including tip, which is based off the subtotal. The tip percentage should be a uniform value across you and your friends.

Note that the final output must consist of **one** of the following types and nothing more:

- If there are less than 0 birthdays, it must cout
  `Number of birthdays is invalid`
- If you enter a cost that is less than 0, it must cout
  `Subtotal is invalid`
- If you enter a tip that is less than 0, it must treat it as 0
- If none of the above conditions hold, it must cout
  `The group needs to pay $(amount)`
  - `(amount)` is whatever you computed the total amount to be

The final output must come after all inputs are typed. Anything else that is not listed in the specifications above is up to you.

This page is intentionally left blank.

This page is intentionally left blank.