Create a Student class with the following private members:

```
    first_ (type string)
    last_ (type string)
    gpa_ (type float)
    id_ (type int)
```

and the following public methods:

```
    full_name() const: string
    id() const: int
    gpa() const: float
    print_info() const: void
    // print_info() should print in the format (GPA has 2 decimal places):
    // Bob Smith, GPA: 3.50, ID: 20146
```

Use an initializer list in the Student constructor.

Write a function that returns a vector of failing students (those whose gpa is < 1.0). Here is part of the implementation:

```
vector<Student> find_failing_students(const vector<Student> &students) {
    vector<Student> failing_students;

    // Always make sure you use the pre-increment operator when working with
    // iterators. Declare iterator inside loop to reduce its scope.
    for (vector<Student>::const_iterator it = students.begin();
            it != students.end(); ++it) {
        // TODO
        // Use .push_back(*it) to add a student to the vector if his or her gpa is
        // less than 1.0.
        // Remember 'it' is a pointer, so in order to get the gpa, you must use '->'
        // instead of '.'
    }

    return failing_students;
}
```

Write a function that prints all the students in the supplied vector:

```
void print_students(const vector<Student> &students) {

    // Same way we iterated through the vector in the previous function, except
    // calling print_info() for each student
}
```

Here's the main:

```
/**
 * Allows the user to enter information for multiple students, then
 * find those students whose GPA is below 1.0 and prints them to the
 * screen.
 */
int main() {
    string first_name, last_name;
    float gpa;
    int id;
    char repeat;
    vector<Student> students;

    do {
        cout << "Enter student's first name: ";
        cin >> first_name;
        cout << "Enter student's last name: ";
        cin >> last_name;
        gpa = -1;
        while (gpa < 0 || gpa > 4) {
            cout << "Enter student's GPA (0.0-4.0): ";
            cin >> gpa;
        }
        cout << "Enter student's ID: ";
        cin >> id;
        students.push_back(Student(first_name, last_name, gpa, id));
        cout << "Add another student to database (Y/N)? ";
        cin >> repeat;
    } while (repeat == 'Y' || repeat == 'y');

    cout << endl << "All students:" << endl;
    print_students(students);

    cout << endl << "Failing students:";
    // TODO
    // Print a space and the word 'None' on the same line if no students are
    // failing.
    // Otherwise, print each failing student on a separate line.

    return 0;
}
```

Zip your makefile and student.cpp source file together in student.zip and upload to Canvas.