

Danh sách các prompt & các câu trả lời tương ứng của ChatGPT

1. Prompt 1

Câu hỏi: Phân tích đề bài



You

Phát sinh danh sách MSSV ngẫu nhiên theo tỉ lệ đã cấu hình: Dựa vào tỉ lệ phát biểu của mỗi sinh viên, phát sinh danh sách MSSV ngẫu nhiên sao cho các sinh viên ít phát biểu sẽ được chọn nhiều hơn trong một session chạy của chương trình.

1 ss chạy là 1 lần bấm run chương trình hả

Câu trả lời:



ChatGPT

Chính xác, một "session" chạy của chương trình có thể được hiểu là một lần bấm run chương trình.

Để phát sinh danh sách MSSV ngẫu nhiên theo tỉ lệ đã cấu hình, bạn có thể sử dụng phân phối ngẫu nhiên dựa trên tỉ lệ của mỗi sinh viên. Dưới đây là một phương pháp bạn có thể sử dụng:

1. **Tính tổng số lần phát biểu của tất cả sinh viên.**
2. **Tính tỉ lệ phát biểu của mỗi sinh viên:** Chia số lần phát biểu của mỗi sinh viên cho tổng số lần phát biểu của tất cả sinh viên.
3. **Tạo một danh sách tương ứng với tỉ lệ phát biểu:** Sử dụng danh sách này để phát sinh MSSV ngẫu nhiên, trong đó mỗi MSSV xuất hiện với xác suất tương ứng với tỉ lệ phát biểu của sinh viên đó.
4. **Chọn ngẫu nhiên MSSV từ danh sách đã tạo:** Sử dụng một hàm ngẫu nhiên để chọn MSSV từ danh sách đã tạo. Đảm bảo mỗi sinh viên chỉ được chọn một lần.

Bằng cách này, các sinh viên ít phát biểu sẽ có xác suất cao hơn để được chọn trong một session chạy của chương trình.

2. Prompt 2

Câu hỏi: Tổ chức fileconfig



You

ID, số lần phát biểu

1720032,1

1720047,1

18200028,1

18200052,1

18200088,1

18200223,1

19200060,2

19200304,2

19200413,2

làm sao để tính được ví dụ: sinh viên có 0 lần lên bảng: tỷ lệ 50%, 1-2 lần: tỷ lệ 25%,.....

Câu trả lời:



ChatGPT

Để tính tỷ lệ phát biểu của sinh viên dựa trên số lần phát biểu, bạn có thể sử dụng thông tin trong file CSV đã cung cấp. Dưới đây là cách tính tỷ lệ theo yêu cầu của bạn:

1. Tính tổng số lần phát biểu của tất cả sinh viên.

2. Tính tỷ lệ phát biểu cho mỗi số lần phát biểu:

- Để tính tỷ lệ cho mỗi số lần phát biểu, bạn cần chia số lượng sinh viên có số lần phát biểu tương ứng cho tổng số sinh viên.
- Ví dụ: Đối với số lần phát biểu là 0, bạn tính tỷ lệ bằng cách chia số lượng sinh viên có 0 lần phát biểu cho tổng số sinh viên.

3. Ghi nhận các tỷ lệ vào file config.txt hoặc sử dụng trực tiếp trong chương trình.

3. Prompt 3

Câu hỏi: Tôi tổ chức như này đúng chứ

```
using namespace std;

class Student
{
private:
    string ID;
    int count;
    float prob;
public:
    Student();
    Student(string _id, int _count, float _prob);
    void setID(string _id);
    void setCount(int _count);
    void setProbability(float _prob);
```

```
    string getID();
    int getCount();
    float getProbability();
};

class Session
{
private:
    vector<Student> student;
public:
    Session() {}
    Session(vector<Student> _student);

    void readFromCSV(string filename);
};
tôi tổ chức như này đúng chứ
```



ChatGPT

Câu trả lời:



ChatGPT

Cấu trúc của bạn là hợp lý và có thể hoạt động tốt. Dưới đây là một số điểm để xem xét:

1. Lớp Student:

- Có các thuộc tính ID, count, và prob để đại diện cho thông tin về sinh viên, số lần phát biểu, và xác suất được gọi phát biểu.
- Có các phương thức set và get để truy cập và thiết lập giá trị cho các thuộc tính.

2. Lớp Session:

- Lớp này chứa một vector của các đối tượng Student, đại diện cho danh sách sinh viên tham gia vào một buổi học.
- Có một constructor mặc định và một constructor nhận một vector của Student để khởi tạo danh sách sinh viên.

3. Phương thức readFromCSV:

- Phương thức này dùng để đọc dữ liệu từ một file CSV và tạo các đối tượng Student tương ứng.
- Điều này sẽ giúp bạn nhập dữ liệu từ file CSV và lưu trữ chúng vào vector student trong lớp Session.

Tuy nhiên, cần lưu ý rằng cách tổ chức này phụ thuộc vào yêu cầu cụ thể của ứng dụng của bạn. Nếu nó đáp ứng được yêu cầu của bạn và dễ dàng sử dụng, thì đó là cách tổ chức đúng cho bạn.

4. Prompt 4

Câu hỏi: Đọc file



You

làm sao khi đọc xong muốn đọc lần 2 tiếp thì đưa con trỏ về đầu kiểu rgif

Câu trả lời:

Trong mã này, sau khi đọc file lần đầu và thêm dữ liệu vào ``set``, chúng ta sử dụng ``file.clear()`` để xóa trạng thái lỗi của file (nếu có), và ``file.seekg(0, ios::beg)`` để đưa con trỏ về đầu file. Sau đó, bạn có thể tiếp tục đọc file từ đầu bằng cách sử dụng ``getline`` hoặc các phương thức đọc khác.



5. Prompt 5

Câu hỏi: Tôi muốn random 1 kiểu dữ liệu bất kì thì làm sao ví dụ như mã số sinh viên

Câu trả lời:

Để tạo ra một kiểu dữ liệu ngẫu nhiên như mã số sinh viên, bạn có thể sử dụng các hàm sinh số ngẫu nhiên có sẵn trong ngôn ngữ lập trình của mình hoặc tạo một hàm để sinh ra mã số ngẫu nhiên theo yêu cầu của bạn. Dưới đây là một ví dụ sử dụng C++:

```
cpp Copy code

#include <iostream>
#include <random>
#include <ctime>

using namespace std;

// Hàm sinh mã số sinh viên ngẫu nhiên
string generateStudentID() {
    string studentID = "SV"; // Tiền tố cho mã số sinh viên
    random_device rd; // Đối tượng sinh số ngẫu nhiên
    mt19937 gen(rd()); // Máy tạo số ngẫu nhiên
    uniform_int_distribution<> dis(1000000, 9999999); // Phạm vi mã số sinh viên
    studentID += to_string(dis(gen)); // Kết hợp tiền tố và số ngẫu nhiên
    return studentID;
}
```

```
int main() {
    // Sử dụng hàm sinh mã số sinh viên ngẫu nhiên
    string studentID = generateStudentID();
    cout << "Student ID: " << studentID << endl;
    return 0;
}
```

Trong ví dụ này, hàm `generateStudentID()` sinh ra một mã số sinh viên ngẫu nhiên bằng cách kết hợp một tiền tố cố định "SV" với một số ngẫu nhiên trong phạm vi từ 1000000 đến 9999999. Bạn có thể điều chỉnh phạm vi này hoặc tiền tố tùy theo yêu cầu của bạn.

6. Prompt 6

Câu hỏi: giải thích hàm Random

```
//Chon 1 hoc sinh bat ki
string Session::chooseRandomStudent()
{
    random_device rd;
    mt19937 gen(rd());
    int minSpeed = randStudent[0].getCount();
    for (int i = 1; i < randStudent.size(); i++)
    {
        minSpeed = min(minSpeed, randStudent[i].getCount());
    }
    vector<Student>result;
    for (int i = 0; i < randStudent.size(); i++)
    {
        if (randStudent[i].getCount() == minSpeed)
        {
            result.push_back(student[i]);
        }
    }
    uniform_int_distribution<> dis(0, result.size() - 1);
    int randomIndex = dis(gen);
    string removeID= result[randomIndex].getID();
    removeStudent(removeID);
    return removeID;
}
```



giải thích hàm random

Câu trả lời:

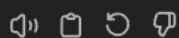


ChatGPT

Hàm `Session::chooseRandomStudent()` được viết để chọn một học sinh ngẫu nhiên từ một danh sách các học sinh có số lần phát biểu ít nhất trong session hiện tại. Dưới đây là giải thích chi tiết về cách hoạt động của hàm:

1. `random_device rd;`: Tạo một đối tượng `random_device` để sinh ra một seed ngẫu nhiên từ các nguồn entropy của hệ thống, như thiết bị ngẫu nhiên.
2. `mt19937 gen(rd());`: Tạo một mô-đun sinh số ngẫu nhiên có tên là `gen` bằng cách sử dụng seed từ `random_device`.
3. `int minSpeed = randStudent[0].getCount();`: Khởi tạo `minSpeed` bằng số lần phát biểu của học sinh đầu tiên trong danh sách `randStudent`.
4. `for (int i = 1; i < randStudent.size(); i++) { ... }`: Duyệt qua danh sách `randStudent` để tìm số lần phát biểu nhỏ nhất (`minSpeed`) trong danh sách.
5. `vector<Student> result;`: Khởi tạo một vector `result` để lưu trữ các học sinh có số lần phát biểu bằng với `minSpeed`.
6. `for (int i = 0; i < randStudent.size(); i++) { ... }`: Duyệt lại qua danh sách `randStudent` để lấy ra các học sinh có số lần phát biểu bằng với `minSpeed` và đưa vào vector `result`.

7. `uniform_int_distribution<> dis(0, result.size() - 1);`: Khởi tạo một phân phối đều trong khoảng từ 0 đến kích thước của `result` trừ đi 1.
8. `int randomIndex = dis(gen);`: Tạo một chỉ số ngẫu nhiên từ phân phối đã khởi tạo, đại diện cho vị trí của học sinh được chọn trong vector `result`.
9. `string removeID = result[randomIndex].getID();`: Lấy ID của học sinh được chọn từ vector `result`.
10. `removeStudent(removeID);`: Gọi hàm `removeStudent` để loại bỏ học sinh đã được chọn ra khỏi danh sách `randStudent`.
11. `return removeID;`: Trả về ID của học sinh đã được chọn ngẫu nhiên.



7. Prompt 7

Câu hỏi: Thiết kế MENU đẹp



You

```
int MENU()
{
    cout<<
```

tạo dùm tôi menu đẹp đẹp với 2 chức năng

1.Random

2.Thoat

kq trả về là lựa chọn

Câu trả lời:

cpp

Copy code

```
#include <iostream>
#include <iomanip>

using namespace std;

int MENU()
{
    cout << "===== " << endl;
    cout << "          MENU          " << endl;
    cout << "===== " << endl;
    cout << "1. Random" << endl;
    cout << "2. Thoat" << endl;
    cout << "===== " << endl;
    cout << "Lua chon cua ban: ";

    int choice;
    cin >> choice;
    return choice;
}
```



```

int main()
{
    int choice;
    do
    {
        choice = MENU();
        switch(choice)
        {
            case 1:
                cout << "Ban da chon Random" << endl;
                // Thêm code của bạn để xử lý chức năng Random ở đây
                break;
            case 2:
                cout << "Cam on ban da su dung chuong trinh!" << endl;
                break;
            default:
                cout << "Lua chon khong hop le. Vui long chon lai!" << endl;
        }
    } while (choice != 2);

    return 0;
}

```

Trong ví dụ này:

- Hàm `MENU()` được sử dụng để hiển thị menu và yêu cầu người dùng nhập lựa chọn.
- Trong `main()`, chúng ta sử dụng một vòng lặp `do-while` để hiển thị menu và xử lý lựa chọn của người dùng.
- Nếu người dùng chọn 1, chúng ta in ra thông báo "Ban da chon Random". Bạn có thể thêm code của mình để xử lý chức năng Random ở đây.
- Nếu người dùng chọn 2, chúng ta in ra thông báo cảm ơn và thoát khỏi chương trình.
- Nếu người dùng nhập một lựa chọn không hợp lệ, chúng ta hiển thị thông báo và yêu cầu người dùng chọn lại.

8. Prompt 8

Trong quá trình giải bài tập Random cũng gặp phải rất nhiều lỗi, phải nhờ đến sự hỗ trợ của ChatGPT để giải quyết nhanh chóng và triệt để.

Câu hỏi: Code + Lỗi

Câu trả lời: Cách giải quyết lỗi trên