

## Hướng dẫn sử dụng bộ test

### 0. Lưu ý trước khi dùng:

Nếu bạn (người dùng) đang tìm một bộ test đúng tuyệt đối, thì bộ test này **không** dành cho bạn. Bộ test này chỉ mang tính chất tương đối, tức đơn thuần là chỉ so sánh kết quả giữa người dùng và kết quả của người viết chương trình (còn được gọi là kì vọng). Việc code của người dùng khác với kì vọng **không** có ý nghĩa khẳng định tính đúng sai mà để biết được cụ thể thì phải đọc test và xem xét kĩ lưỡng mới có thể đưa ra được quyết định.

### 1. Giới thiệu về các file:

Trong tệp tin sẽ có những file sau:

- InputGen.bat là chương trình tạo test, mỗi lần chạy sẽ tạo ngẫu nhiên 1000 test.
- ExpectGen.bat là chương trình sinh ra dự đoán dựa trên 1000 test được cung cấp.
- OutputGen.bat là chương trình chạy code của người dùng và so sánh với kì vọng.

Các tệp tin input, output và expected đều được lưu trong thư mục Tests và các chương trình có thể chạy độc lập với nhau, tuy nhiên sẽ có một chương trình ALL IN ONE.bat có thể chạy cùng một lúc cả 3 chương trình trên.

### 2. Các chỉnh sửa trong code:

Để code có thể sinh ra được output đúng với format được quy ước sẵn, người dùng cần chỉnh sửa code của mình như sau:

- Viết thêm hàm **getChallenge()** cho PassingCard trả về thuộc tính **challenge** của nó và các PassingCard khi in ra giờ đây phải đi kèm với challenge của chúng được để trong dấu ngoặc đơn. Ví dụ: `"PassingCard(RobotS)"`, `"PassingCard(all)"`,...

Trong phần **StudyPinkProgram**:

- Khai báo biến ofs kiểu ofstream có thuộc tính private: `ofstream ofs;`
- Chỉnh sửa constructor của StudyPinkProgram:

```
StudyPinkProgram(const string & config_file_path, const string & outfile){  
    *(your original code)*  
    ofs.open(outfile);  
    ofs << config->str() << "\n\n";  
}
```

- Thay thế các hàm **printResult** và **printStep** thành như sau (nhớ đổi tên các hàm và biến cho phù hợp):

```
void printResult() {  
    if (sherlock->getCurrentPosition().isEqual(criminal->getCurrentPosition())) {
```

```

        ofs << "Sherlock caught the criminal\n";}

else if (watson->getCurrentPosition().isEqual(criminal->getCurrentPosition())) {

    ofs << "Watson caught the criminal\n";}

else {

    ofs << "The criminal escaped\n";}

ofs << "----- Test end -----";

ofs.close();

};

void printStep(int si) {

    ofs << "*** Step: " + to_string(si) + " **\n";

    ofs << "-- Sherlock: " + sherlock->str() + "\n";

    ofs << "Stats: " + to_string(sherlock->getHp()) + " HP ~ " + to_string(sherlock->getExp()) + " EXP\n";

    ofs << "Sherlock Bag: " + sb->str() + "\n";

    ofs << "-- Watson: " + watson->str() + "\n";

    ofs << "Stats: " + to_string(watson->getHp()) + " HP ~ " + to_string(watson->getExp()) + " EXP\n";

    ofs << "Watson Bag: " + wb->str() + "\n";

    ofs << "-- Criminal: " + criminal->str() + "\n";

    ofs << "-- Robots: " + to_string(arr_mv_objs->size()-3) + "\n";

    for (int i=3; i<arr_mv_objs->size(); i++)

        {   ofs << arr_mv_objs->get(i)->str() + " ~ item: ";

            switch (arr_mv_objs->get(i)->getItem()->gettype()){

                case (MAGIC_BOOK):

                    ofs << "MagicBook";

                    break;

                case (ENERGY_DRINK):

                    ofs << "EnergyDrink";

                    break;

                case (FIRST_AID):

                    ofs << "FirstAid";

                    break;

                case (EXCEMPTION_CARD):

```

```

        ofs << "ExemptionCard";

        break;

    case (PASSING_CARD):

        ofs << "PassingCard(" << arr_mv_objs->get(i)->getItem()->getChallenge() << ")";

        break;

    default:

        ofs << "UNDEFINED";

        break;

    }

    ofs << "\n";

}

ofs << "\n";

};

```

### 3. Cách chạy và kiểm tra:

Sau khi hoàn thành các bước trên, để chạy chương trình, người dùng cần copy các file **study\_in\_pink2.h** và **study\_in\_pink2.cpp** vào trong thư mục chứa các file này, ghi đè các file có sẵn (không cần ghi đè file **main.h** vì đó là file được cung cấp trên lms). Đến đây, người dùng chỉ cần chạy các file đã được mô tả ở mục 1 theo ý muốn và so sánh với kì vọng.

Có thể so sánh sự khác nhau tại <https://www.diffchecker.com/text-compare/>. Một trong những cách để kiểm tra tính đúng đắn đó là so sánh ở step lớn nhất mà output và expected giống nhau rồi chạy thủ công các step kế tiếp.