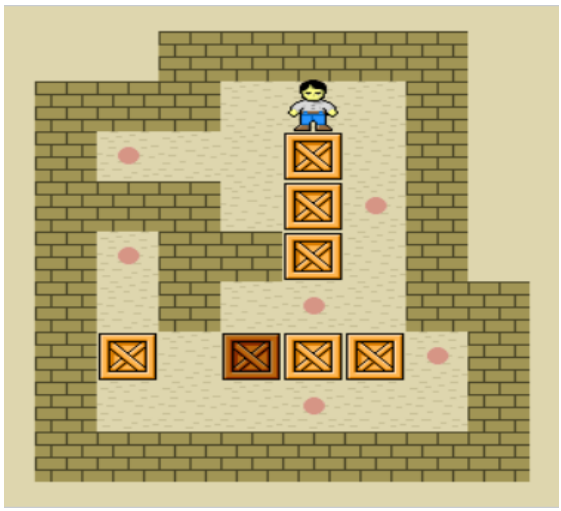
BÀI TẬP LỚN

# TRÒ CHƠI SOKOBAN

## I. Giới thiệu

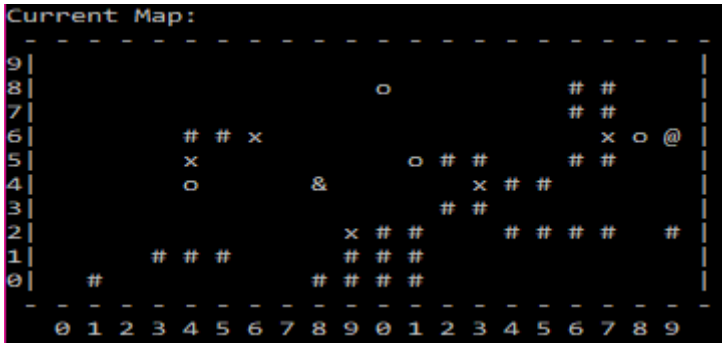
Sokoban là trò chơi puzzle nổi tiếng được phát minh vào năm 1981 bởi Hiroyuki Imabayashi. Trong trò chơi Sokoban, người chơi cần đẩy các thùng gỗ trong một nhà kho có nhiều bức tường ngăn cách đến các mục tiêu cho trước (Hình 1). Lấy ý tưởng từ trò chơi sokoban, trong bài tập lớn này sinh viên cần hiện thực chương trình hoàn thành các bước đi được cho trước của người chơi trong trò chơi Sokoban.



Hình 1: Trò chơi Sokoban

## II. Đặc tả chương trình

### 1. Bản đồ trò chơi




Hình 2: Bản đồ trò chơi của chương trình

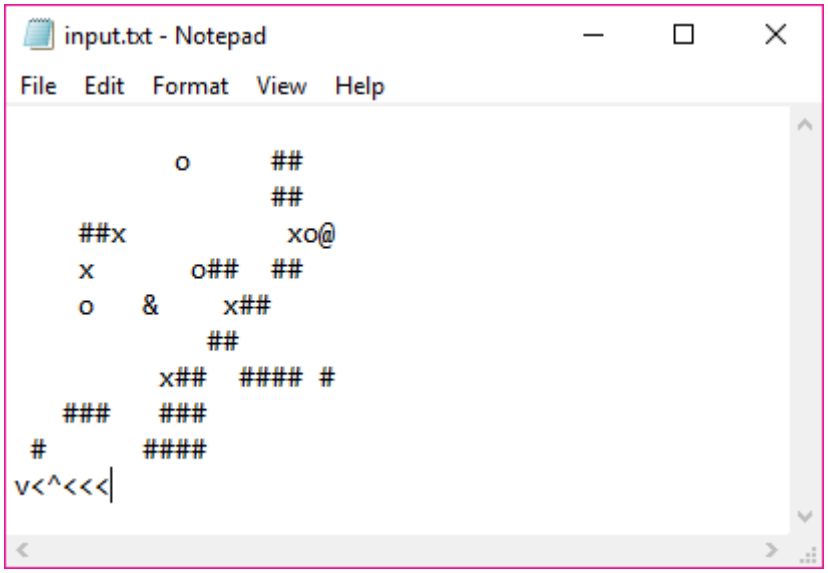
Bản đồ trò chơi được biểu diễn bằng một mảng kí tự hai chiều có tối đa 10 dòng và 20 cột. Mỗi một vị trí trong bản đồ thuộc một trong các phạm trù sau:

- Kí tự '@': Thể hiện vị trí của người chơi trong bản đồ, chỉ có duy nhất một người chơi trong bản đồ.
- Kí tự 'o': Thể hiện vị trí các thùng gỗ trong bản đồ, có tối đa 4 thùng gỗ trong trò chơi.
- Kí tự 'x': Thể hiện vị trí các mục tiêu mà người chơi cần đẩy các thùng gỗ tới trong bản đồ. Số ô mục tiêu x luôn luôn bằng số thùng gỗ o cộng thêm 1.
- Kí tự ' ': Thể hiện vị trí các khoảng trống trong bản đồ.
- Kí tự '#': Thể hiện vị trí các bức tường ngăn cách không cho người chơi đi qua.

Bản đồ trò chơi sẽ được đọc từ file input. Sinh viên có thể dùng hàm **printCurrentMap** để in bản đồ ra console như Hình 2.

## 2. Dữ liệu đầu vào

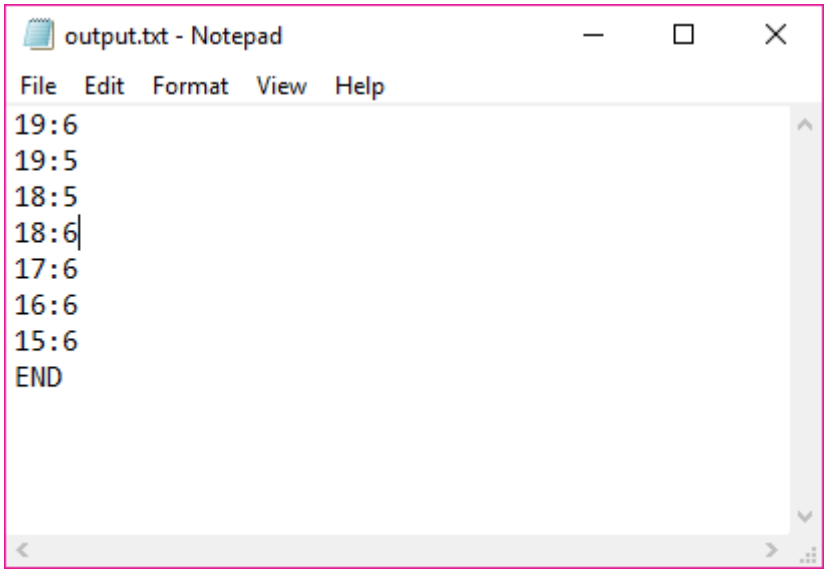
Dữ liệu đầu vào là một file .txt chứa **thông tin bản đồ** và **danh sách các bước đi** người chơi cần thực hiện. Thông tin bản đồ được thể hiện ở 10 dòng đầu, mỗi dòng bao gồm 20 kí tự. Danh sách các bước đi được thể hiện ở dòng cuối cùng. Hình 3 là một ví dụ hợp lệ của file input cho chương trình. Sinh viên có thể thay đổi file input để test chương trình.




Hình 3: File input chứa dữ liệu đầu vào

## 3. Dữ liệu đầu ra

Chương trình sẽ được thực thi dựa vào danh sách các bước đi đọc từ file input. Sinh viên cần in ra file output vị trí của người chơi sau mỗi bước đi. Sinh viên có thể dùng hàm **writeStep** để xuất vị trí ra file output.




Hình 4: File output của chương trình

Khi chương trình kết thúc, kết quả của trò chơi sẽ được xuất ra file output.txt, có ba kết quả là WIN, LOSE hoặc END. Hình 4 là output của chương trình khi chạy file input ở Hình 3.

### 4. Cách di chuyển của người chơi

Trong trò chơi này, người chơi chỉ được phép di chuyển qua các ô trống hoặc các ô mục tiêu. Người chơi không thể đi xuyên qua tường cũng như vượt quá giới hạn của bản đồ. Người chơi chỉ được phép đi lên trên, xuống dưới, sang trái hoặc sang phải. Các bước đi của người chơi được cho trước, vì vậy người chơi sẽ di chuyển dựa theo các bước đi đọc từ file input như sau:

- Nếu bước đi là '^', người chơi di chuyển lên trên.
- Nếu bước đi là 'v', người chơi di chuyển xuống dưới.
- Nếu bước đi là '<', người chơi di chuyển sang trái.
- Nếu bước đi là '>', người chơi di chuyển sang phải.

**Ví dụ:** Trong bản đồ ở Hình 5, người chơi đang ở vị trí (16, 6). Lúc này người chơi có thể di chuyển sang trái hoặc sang phải nhưng không thể di chuyển lên trên hay xuống dưới.


Hình 5: Người chơi không thể di chuyển lên trên hoặc xuống dưới vì bị ngăn bởi các bức tường

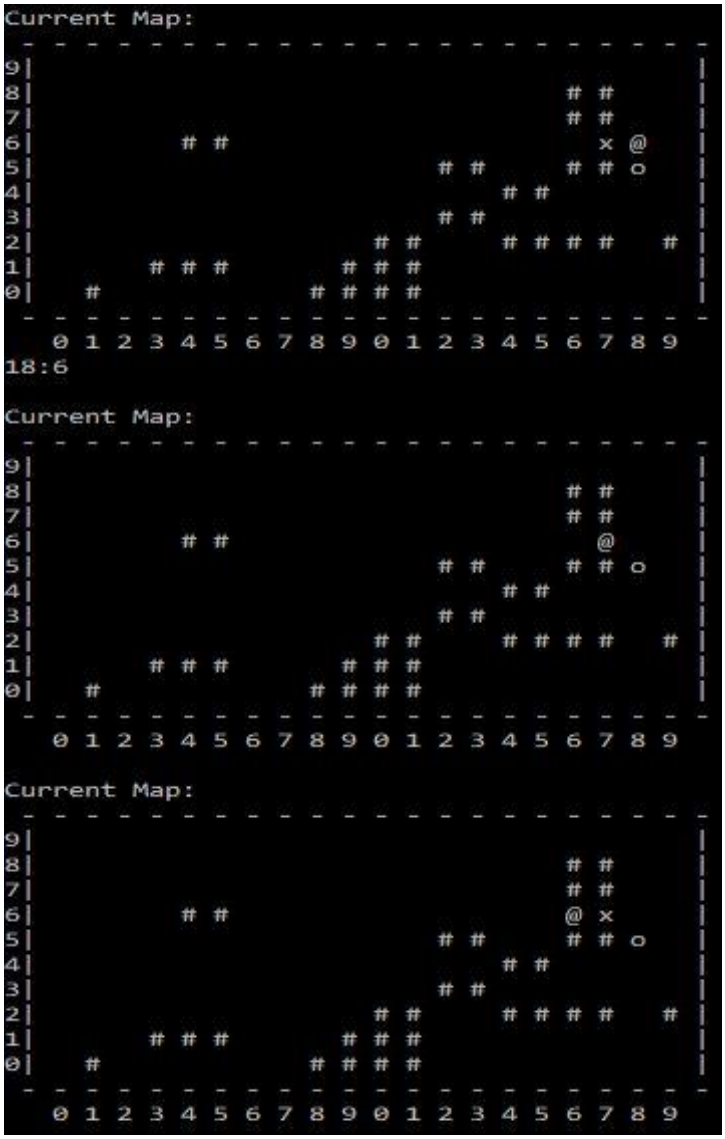
Nếu bước đi hiện tại là '<', người chơi sẽ di chuyển sang trái. Sau bước đi này, người chơi sẽ ở vị trí mới (15, 6) như Hình 6.


Hình 6. Người chơi di chuyển sang trái

Nếu bước đi làm người chơi đụng tường hoặc vượt quá giới hạn bản đồ, bước đi đó sẽ không có hiệu lực, người chơi giữ nguyên vị trị hiện tại. Chẳng hạn, khi người chơi ở vị trí (16, 6) trong bản đồ Hình 5, nếu bước đi hiện tại là '^', vị trí mới của người chơi vẫn là (16, 6).

Người chơi có thể di chuyển qua ô mục tiêu.

**Ví dụ:** Người chơi di chuyển qua mục tiêu bằng cách di chuyển sang trái 2 lần được thể hiện ở Hình 7.




Hình 7: Người chơi đi qua ô mục tiêu

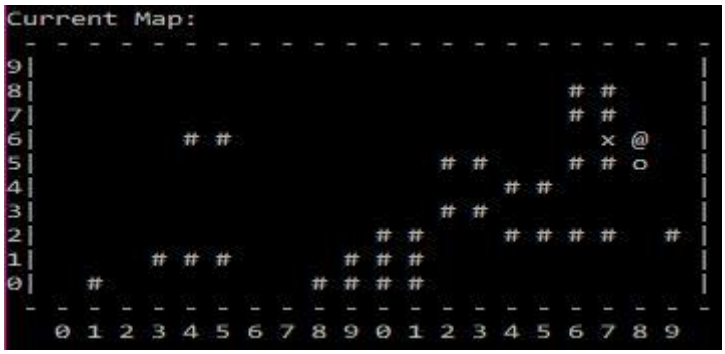
### 5. Đẩy thùng gỗ

Nếu người chơi di chuyển tới vị trí thùng gỗ, người chơi sẽ đẩy thùng gỗ theo hướng đi hiện tại của người chơi. Người chơi chỉ có thể đẩy thùng gỗ sang trái, phải, trên, dưới và chỉ đẩy tới các ô trống hoặc ô mục tiêu. Người chơi không thể đẩy thùng gỗ xuyên tường hoặc vượt quá giới hạn bản đồ.

**Ví dụ:** Trong bản đồ Hình 8, người chơi đang ở vị trí (18, 7), thùng gỗ ở vị trí (18, 6). Nếu bước đi hiện tại là ‘v’, người chơi sẽ đẩy thùng gỗ xuống dưới.


Hình 8: Bản đồ trước khi người chơi đẩy thùng gỗ

Sau bước đi ‘v’, vị trí mới của người chơi là (18, 6) và của thùng gỗ là (18, 5). Bản đồ mới được cập nhật như Hình 9.

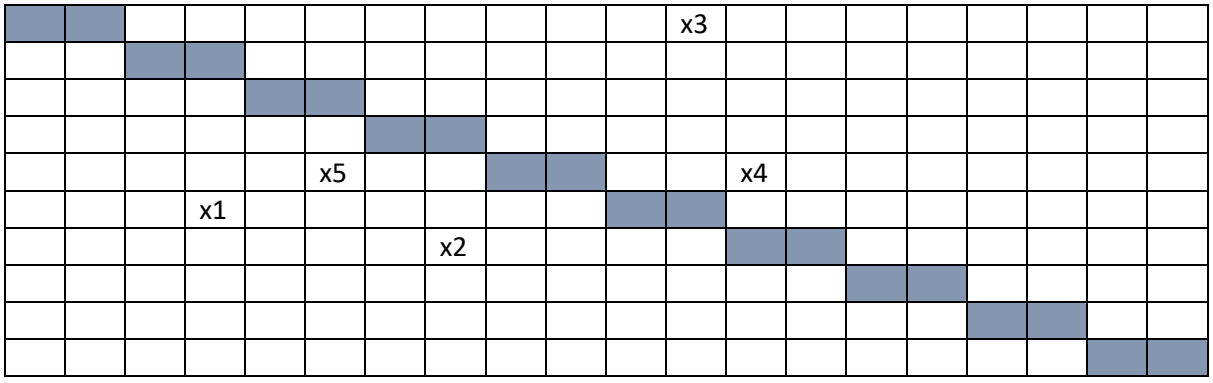



Hình 9: Bản đồ sau khi người chơi đẩy thùng gỗ

### 6. Ô mục tiêu

Số ô mục tiêu x được cho trước bằng số thùng gỗ cộng thêm 1. Trong số các mục tiêu này, có một mục tiêu là giả. Người chơi không thắng trò chơi nếu đẩy thùng gỗ tới mục tiêu giả. Mục tiêu giả được xác định có tổng các khoảng cách theo chiều dọc và chiều ngang tới đường chéo là ngắn nhất. Trong trường hợp có nhiều mục tiêu thỏa mãn điều kiện này, mục tiêu có tổng hoành độ và tung độ nhỏ nhất là mục tiêu giả.

**Ví dụ:** Theo bảng dưới, các mục tiêu x4, x5 có tổng các khoảng cách tới đường chéo ngắn nhất (d=3) nhưng mục tiêu x5 là giả vì có tổng các tọa độ nhỏ hơn mục tiêu x4.

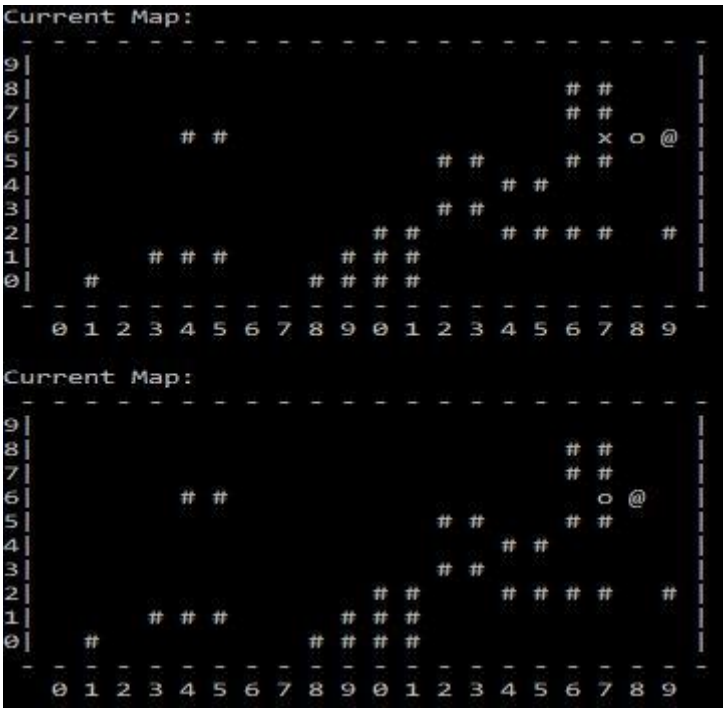


### 7. Trạng thái kết thúc của trò chơi

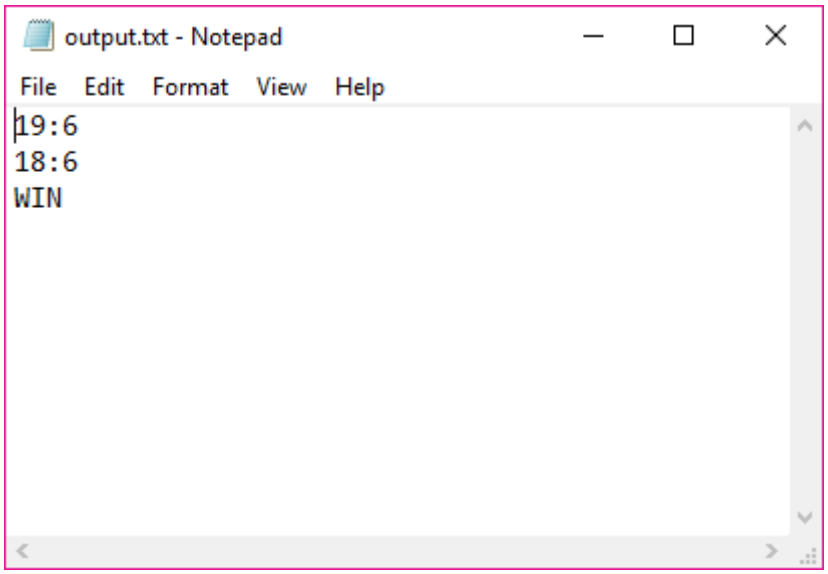
a. Trạng thái WIN

Nếu người chơi đẩy được tất cả các thùng gỗ tới ô mục tiêu, chương trình xuất ra kết quả của trò chơi là WIN và kết thúc. Các bước đi tiếp theo (nếu còn) sẽ không được thực thi.

**Ví dụ:** Giả sử ban đầu người chơi ở vị trí (19, 6), danh sách các bước đi là "<<<". Người chơi cần di chuyển sang trái 3 lần. Sau bước đi đầu tiên, người chơi đã đẩy thùng gỗ sang trái tới mục tiêu (Hình 10). Chương trình xuất ra file output kết quả trò chơi là WIN (Hình 11).




Hình 10: Người chơi đẩy thùng gỗ về mục tiêu

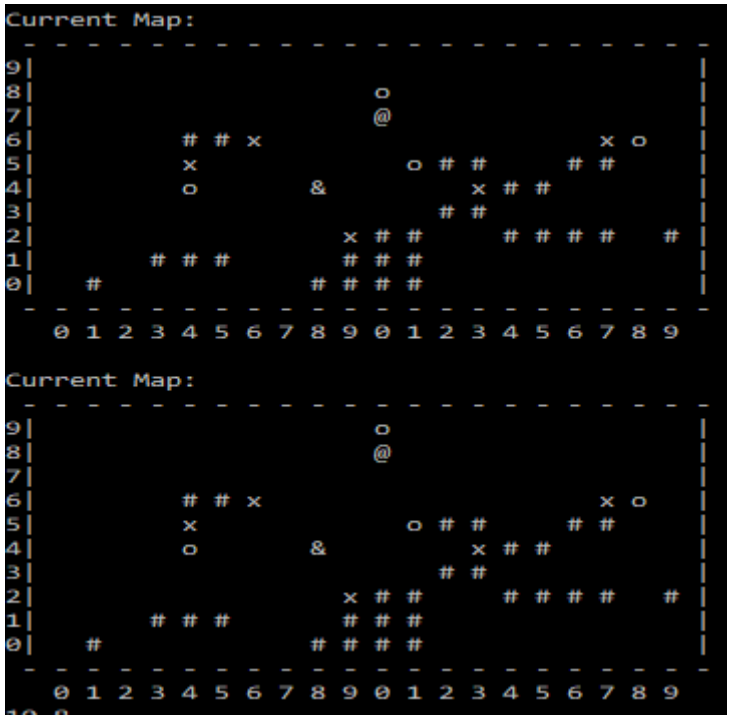


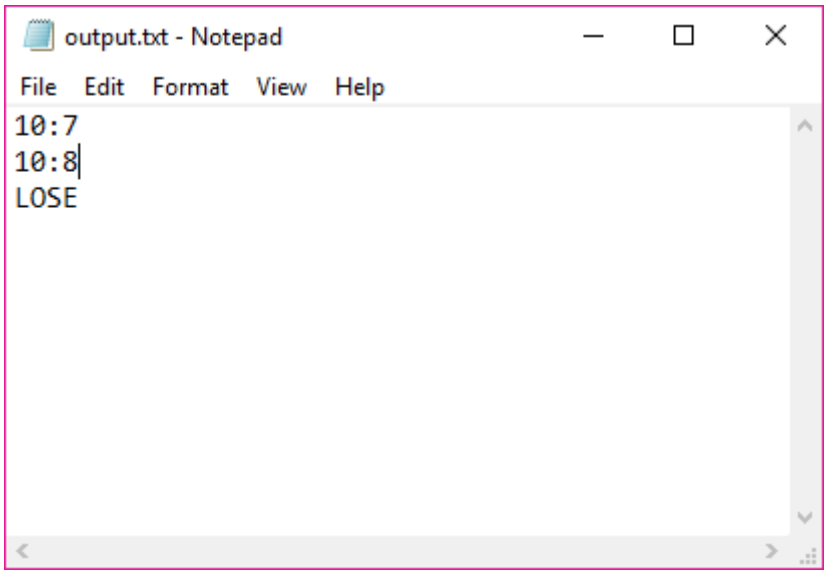

Hình 11: Chương trình xuất kết quả WIN ra file output

b. Trạng thái LOSE

Chương trình xuất ra kết quả của trò chơi là LOSE và kết thúc ngay khi phát hiện người chơi không thể đẩy thùng gỗ tới ô mục tiêu. Sinh viên cần phát hiện được các trường hợp thùng gỗ ở "vị trí chết".

**Ví dụ:** Giả sử ban đầu người chơi ở vị trí (10, 7), danh sách các bước đi là "^>>>". Sau bước đi đầu tiên, người chơi đẩy thùng gỗ lên trên. Lúc này thùng gỗ ở vị trí mới là (10, 9). Nhận thấy không thể đẩy thùng gỗ đến ô mục tiêu (vì không có cách nào đẩy thùng gỗ ra khỏi đường biên của bản đồ), chương trình xuất ra file output với trạng thái LOSE và kết thúc.




Hình 12: Người chơi đẩy thùng gỗ vào "vị trí chết" nên chương trình xuất kết quả LOSE ra file INPUT

c. Trạng thái END

Sau khi hoàn thành tất cả các bước đi, nếu vẫn còn thùng gỗ chưa được đẩy tới ô mục tiêu. Chương trình xuất ra trạng thái END và kết thúc.

## III. Hướng dẫn thực hiện

- ✓ Sinh viên chỉ được code trong file **assignment1.c**, mục **TODO**.
- ✓ Sinh viên có thể tự viết các hàm con khác để sử dụng trong chương trình.
- ✓ Sinh viên có thể thay đổi file **input.txt** để kiểm tra chương trình.
- ✓ Sinh viên không được include thêm bất kì thư viện nào khác.

## IV. Chấm điểm

Bài tập lớn 1 được biên dịch và chấm trên Code::Blocks version 16.01. Sinh viên cần kiểm tra chương trình chạy thành công trên Code::Blocks trước khi submit. Nếu chương trình của sinh viên bị lỗi biên dịch khi chấm do không tương thích môi trường, sinh viên sẽ bị 0 điểm cho bài tập lớn này**.**

## V. Hình phạt

Sinh viên **không được copy code lẫn nhau**. Trong trường hợp có gian lận phát hiện, tất cả các sinh viên liên quan đều bị 0 điểm.

Sinh viên **không được include thêm bất kì thư viện nào khác**, nếu không sinh viên sẽ bị 0 điểm.

## VI. Thời gian nộp bài

Sinh viên chỉ cần submit file assignment1.c lên elearning. Nếu submit các file khác hoặc file không đúng tên, sinh viên sẽ bị 0 điểm.

Thời hạn của bài tập lớn 1 là **23h55** ngày **03/12/2017.**