**TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI   
THÀNH PHỐ HỒ CHÍ MINH**

ĐỀ TÀI: ROBOT DI CHUYỂN TRÊN MA TRẬN

Giáo viên hướng dẫn : Bùi Thị Thà

Môn : Kĩ thuật lập trình

Sinh viên thực hiện : Nguyễn Văn Huy

Lớp : CN22D

ĐOẠN CODE XỬ LÝ ĐƯỢC CÁC YÊU CẦU CỦA ĐỀ BÀI :

Robot chỉ có thể đi theo 4 hướng(trên, dưới, trái, phải). Robot sẽ lựa chọn hướng(ô) có giá trị lớn để đi, các ô đi rồi sẽkhông đi lại.Điểm được tính bằng tổng giá trị các ô robot đi qua.

1. Áp dụng các kiến thức trong môn kỹ thuật lập trình đệ quy/struct/ cấp phát động/ đọc ghi file

2. Cài đặt thuật toán xuất được kết quả.

3. Trường hợp đặt 2 robot ở 2 vị trí khác nhau. Xuất kết quả 2 robot, so sánh kết quả, xuất những vị trí 2 robot đi trùng ô nhau.

4. Cho phép 2 người chơi đặt 2 robot ở vị trí bất kỳ trên ma trận, mỗi lượt chỉ được 1 robot di chuyển, các ô robot đi rồi, robot khác không được đi lại. Xuất kết quả của 2 robot

5. Bài toán mở, sinh viên có mở rộng, thêm 1 số tính năng sáng tạo.Ví dụ: visualize đường đi robot, mô phỏng step by step,….

HÀM XỬ LÝ CỦA CHƯƠNG TRÌNH :

1. **void read\_file**

Đọc dữ liệu từ file input.txt, in ra ma trận đầu tiên và gán dữ liệu vào mảng và các biến

1. **int tim\_duong\_1robot**

tìm đường đi cho robot ở cả trường hợp 1 robot và 2 robot di chuyển độc lập

1. **bool tim\_duong\_2robot\_2**

tìm đường đi cho trường hợp mỗi lượt chỉ được 1 robot di chuyển

1. **void write\_output1**

ghi dữ liệu kết quả vào file output.txt cho trường hợp 1 robot di chuyển

1. **void write\_output**

ghi dữ liệu kết quả vào file output.txt cho trường hợp 2 robot di chuyển

1. **một số hàm tính toán khác**

* int score
* void trung\_nhau
* string robot\_win

1. **int main**

xử lý theo cấu trúc của chương trình.

**KHAI BÁO THƯ VIỆN**

* <fstream> để nhập, ghi dữ liệu từ file bên ngoài..
* <chrono> <thread> thời gian xử lý code giữa các bước di chuyển.
* <iomanip> tạo khoảng cách ma trận, tạo màu.

1. **void read\_file**

* gán số hàng, số cột cho rows, cols và giá trị từng phần tử trên ma trận vào mảng 2 chiều maze[][].
* in ra ma trận ban đầu trong file input.txt.

1. **int tim\_duong\_1robot**

* sử dụng đệ quy để tìm đường đi, đến khi không thỏa điều kiện if1 thì ngưng cập nhật vị trí và trả về số bước nhảy của robot (path\_size), cập nhật giá trị từng bước nhảy (path[]).
* visited[][] để lưu vị trí đã đi qua bằng true.

1. **bool tim\_duong\_2robot\_2**

* mục đích dùng bool trả về giá trị true hoặc false để bước nhảy của robot trở nên tối ưu hơn cho trường hợp mỗi lượt 1 robot di chuyển.

1. **void write\_output1**

* ghi giá trị vào file output.txt

1. **void write\_output**

* ghi các giá trị vào file output.txt

tìm các vị trí trùng nhau và ghi file output.txt

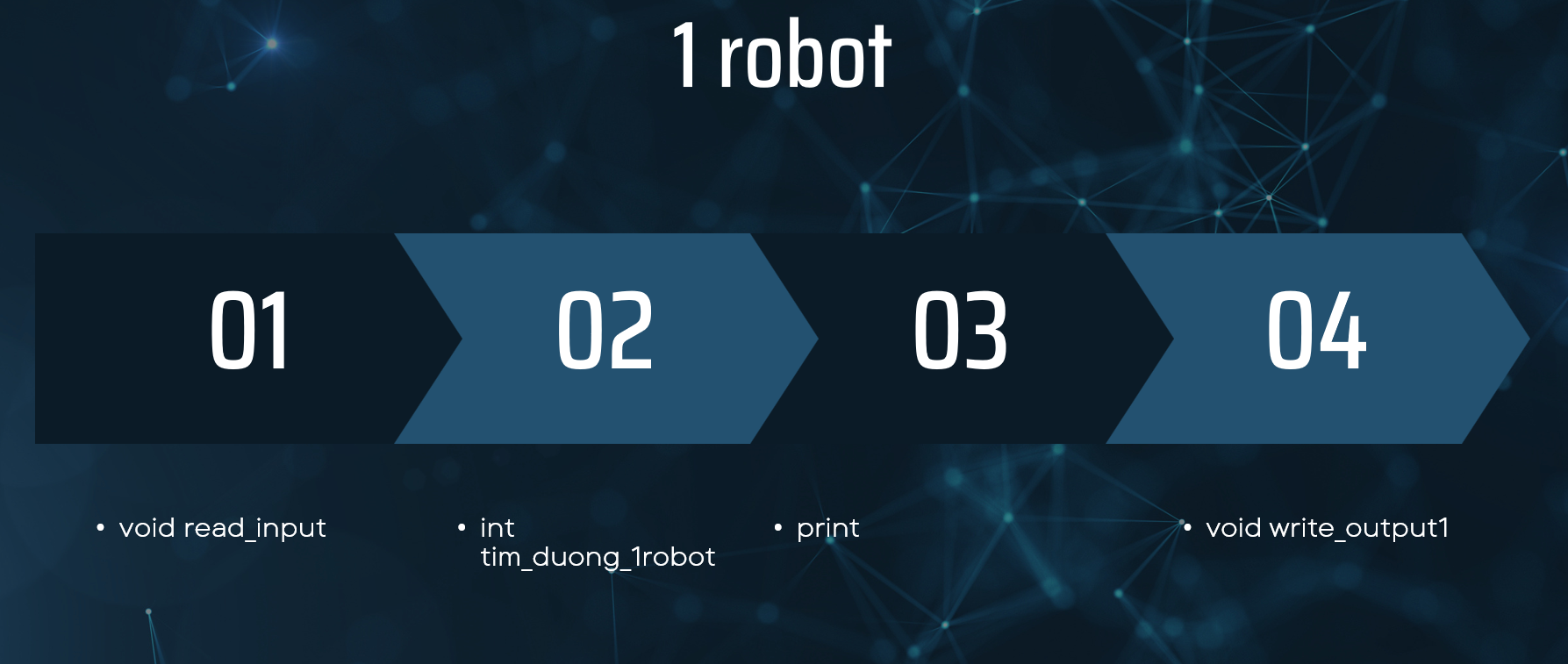
1. **Một số hàm tính toán khác**

* **Int score (hàm tính điểm của robot )**
* sử dụng để tính điểm cho cả 3 trường hợp di chuyển của robot.
* sử dụng số lần di chuyển của robot và giá trị vị trí di chuyển cộng lại để tính điểm.
* **void trung\_nhau ( hàm tìm giá trị đường đi trùng nhau của 2 robot )**
* sử dụng cho trường hợp 2 robot di chuyển trùng nhau và in vị trí trùng.
* **string robot\_win ( in ra kết quả của 2 robot )**
* tạo kết quả của trò chơi bằng việc so sánh điểm của 2 robot và trả về :
* "ROBOT 1 WIN" || "ROBOT 2 WIN" || "ROBOTS TIE"

1. **int main**

* **sử dụng cho 1 robot di chuyển**
* in ra số bước di chuyển, điểm số và những vị trí đã di chuyển qua.
* sử dụng hàm score để tính điểm.
* ghi giá trị vào file input.
* **sử dụng cho 2 robot di chuyển**
* tương tự như di chuyển 1 robot , sau khi robot chạy ta thêm 1 vòng lặp để reset visited[][] về false để xử lý tiếp robot 2 và xuất ra kết quả.
* sử dụng hàm trung\_nhau để in ra vị trí trùng nhau.
* sử dụng hàm robot\_win() để in ra robot chiến thắng
* dùng write\_input để ghi vào file output.txt
* **sử dụng cho TH không đi qua vị trí robot khác đã đi qua.**
* sử dụng một biến bool robot\_1\_turn để mỗi lượt chỉ một robot di chuyển bằng cách khai báo robot\_1\_turn = true, khi robot\_1\_turn = true robot 1 sẽ di chuyển, chuyển robot\_1\_turn = false , robot 2 di chuyển. Sử dụng hàm find\_next\_move () để tìm vị trí tiếp theo cho robot.
* in các kết quả ra ngoài màn hình
* ghi vào file output.txt

**Quy trình xử lý thuật toán**

****

A screenshot of a computer

Description automatically generated

THE END