

## HƯỚNG DẪN CÀI ĐẶT TRẠNG THÁI VÀ CÁC HÀNH ĐỘNG

Để lập trình các bài toán tìm kiếm, học viên cần phải định nghĩa được trạng thái và các hành động

### Bước 1. Định nghĩa trạng thái

- Dùng struct để khai báo cấu trúc trạng thái
- Khai báo trạng thái bắt đầu
- Xây dựng hàm in trạng thái
- In trạng thái ra màn hình để kiểm tra khai báo cấu trúc và hoạt động của hàm in

Ví dụ: mã nguồn định nghĩa trạng thái bài toán đong nước

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  // Khai bao cau truc trang thai
5  typedef struct {
6      int x; // Luong nuoc trong binh x
7      int y; // Luong nuoc trong binh y
8  } State;
9
10 // Ham in trang thai
11 void print_State(State state) {
12     printf("\n  X: %d --- Y:%d", state.x, state.y);
13 }
14
15 int main() {
16     State cur_state = {5, 4};
17     State result;
18
19     printf("Trang thai bat dau");
20     print_State(cur_state);
21
22     return 0;
23 }
```

## Bước 2. Định nghĩa các hành động

- Định nghĩa từng hành động
- Gọi từng hành động lên kiểm tra

Ví dụ: Mã chương trình định nghĩa hành động làm đầy nước bình X

```
19 // Ham lam day nuoc binh X
20 int pourWalterFullX(State cur_state, State *result) {
21     if(cur_state.x < tankcapacity_X) {
22         result->x = tankcapacity_X;
23         result->y = cur_state.y;
24         return 1;
25     }
26     return 0;
27 }
```

Ví dụ: Mã chương trình kiểm tra hành động làm đầy nước bình X

```
29 int main() {
30
31     State cur_state = {5, 4};
32     State result;
33
34     printf("Trang thai bat dau");
35     print_State(cur_state);
36
37     pourWalterFullX(cur_state, &result);
38     printf("\nTrang thai sau hanh dong");
39     print_State(result);
40
41     return 0;
42 }
43
```