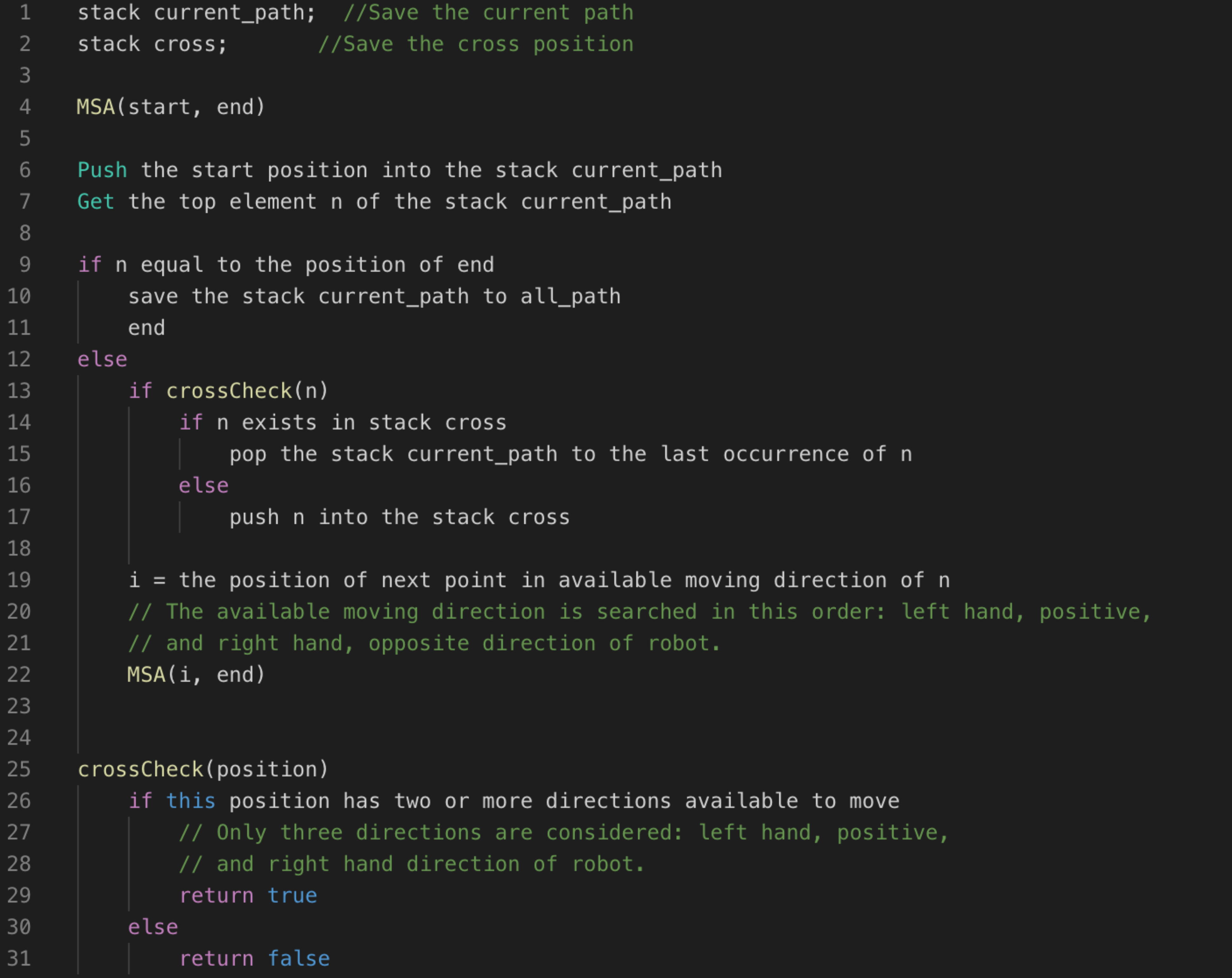

```
stack current_path;  //Save the current path
stack cross;       //Save the cross position

MSA(start, end)

Push the start position into the stack current_path
Get the top element n of the stack current_path

if n equal to the position of end
    save the stack current_path to all_path
    end
else
    if crossCheck(n)
        if n exists in stack cross
            pop the stack current_path to the last occurrence of n
        else
            push n into the stack cross

    i = the position of next point in available moving direction of n
    // The available moving direction is searched in this order: left hand, positive,
    // and right hand, opposite direction of robot.
    MSA(i, end)


crossCheck(position)
    if this position has two or more directions available to move
        // Only three directions are considered: left hand, positive,
        // and right hand direction of robot.
        return true
    else
        return false
```