

# CSE102#HW07

## #part1 40pts

In this part, you will write a **recursive** palindrome finder function. A palindrome is a sequence of characters which reads the same backward or forward (“abcdedcba”, “abba”). This function has some differences from standard palindrome.

- In the input string, there may be punctuations, spaces etc. that you must ignore. **You will match only letters.** For example, “ab, cd!ed~cb%a” is a palindrome but “ab!ab” is not.
- The function must work **case insensitive**. For example, “AbcDedCba” is a palindrome.

Your function returns 1 if the given string is a palindrome, otherwise (any invalid conditions and not palindrome) it returns 0.

### Signature

```
int isPalindrome(char * str);
```

### Sample Usage

```
isPalindrome("ab, cd!ed~cb%a ");
```

### Return Value

1

## #part2 60pts

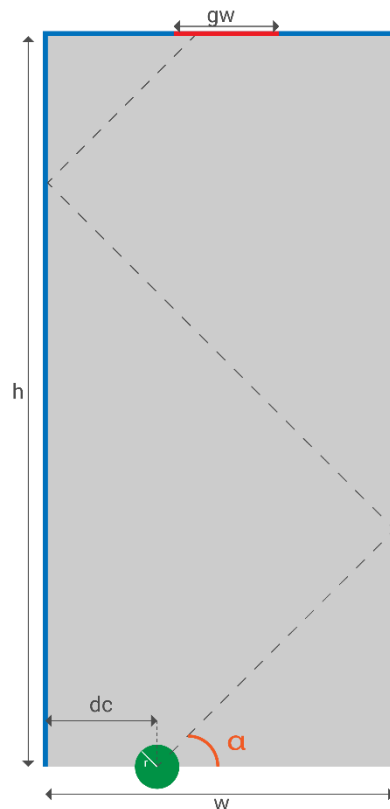


Figure 1: Air hockey table.

In this part, you will write a **recursive** function that simulates hitting an air hockey disc and checks whether the disc will be in the goal or not. Air hockey table is shown in the Figure 1. The disc is always placed on the bottom (closer to the player) end of the table. The goal is on the opposite side of the table and centered.

Function parameters are table width (**w**), table height (**h**), disc center distance from left side of the table (**dc**), kick angle between horizontal axis or kick direction ( **$\alpha$** ), disc radius (**r**), and goal width (**gw**).

Your function should take into account:

- Kick the disc with  $\alpha$  angle ( $\alpha$  can be in between 5 degrees and 175 degrees)
- Handle reflections. Notice that the disc may reflect multiple times from vertical ends (left and right in the figure) of the table before it reaches to the top side.
- Score happens when the disc hits the goal opening (hint: checking the center location will not do).

### Reflection Function

If the disc hits one of the vertical table sides (left or right side), the table side reflects it. The reflection is not a perfect reflection. There will be a random diversion between -5 degrees and +5 degrees. If the disc hits a side with  $\alpha$  angle, the reflection angle must calculate like the disc hits the side with  $(\alpha + \text{randomDiversiion})$  angle. You do not have to write the random number generation function. We will give you `randNumber` function to generate a random integer (**within range [0,MAX RAND]**). The angle parameter for this function is the angle before the disc hits to the table side. **If the reflection angle is bigger than 175 degrees, you will set it to 175 degrees and for smaller than 5 degrees you will set it to 5 degrees.**

### Goal Test

If the disc hits top side of the table, you will check whether the ball is in the goal or not. Goal is at the horizontal center of the top side with a length of gw. If the ball is in the goal `kickDisc` function returns 1, otherwise (any invalid conditions and not in goal condition) it returns 0.

### Random Number Function

```
int randNumber(int angle){
    srand(angle);
    return rand();
}
```

#### Signature

```
int reflection(int angle);
int kickDisc(double wallWidth, double wallHeight, double ballCenterX, double
kickAngle, double ballRadius, double goalWidth);
```

#### Sample Usage

```
kickDisc(10, 26, 2, 45, 1, 2);
```

#### Return Value

1

### #notes

- You should not send your main function. You should send the functions described above and your helper functions if you have any.
- Use the concepts you learned in the class such as loops, conditions and functions whenever needed.
- The assignment must be your original work. Duplicate or very similar assignments are both going to be considered as cheating.
- Ask your questions via moodle.