



Zeid Kootbally
University of Maryland
College Park, MD

zeidk@umd.edu

RWA-1 on Lecture 3: Arrays and Vectors

ENPM809Y : Spring 2020
Due **Wednesday, February 19, 2020**

Contents

Assignment

2

Assignment

- The assignment uses the tray and pegs depicted in Figure 1 as reference.

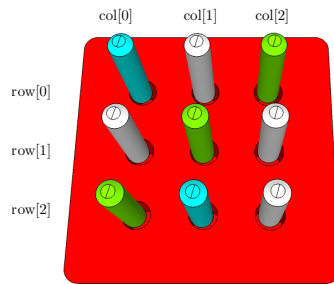


Figure 1: Tray and pegs.

Goal

- The goal of this assignment is to represent the positions of the pegs in the tray using arrays and vectors.

Instructions

- Create a 2D array of string objects with 3 rows and 2 columns.
- Initialize the array to have the following elements:
 - First row: **blue, white** (contains pegs of the first row of the tray)
 - Second row: **green, white** (contains pegs of the second row of the tray)
 - Third row: **green, white** (contains pegs of the third row of the tray)
- Create 3 vectors of string objects where each vector represents a row of pegs:
 - `color_vec1`
 - `color_vec2`
 - `color_vec3`
- Read the 2D array and store values of each row in a vector:
 - `color_vec1`: First row of the array.
 - `color_vec2`: Second row of the array.
 - `color_vec3`: Third row of the array.
- Prompt the user to enter the color for the missing peg using the picture of the tray:
 - The user should enter: **green white blue**
- Appropriately insert the first, second, and third input in `color_vec1`, `color_vec2`, and `color_vec3`, respectively.
 - Make sure to insert these inputs in the correct place in each vector, i.e:
 - green should be placed at the end of `color_vec1` (to match the first row of the tray).

- white should be placed at the beginning of `color_vec2` (to match the second row of the tray).
- blue should be placed between green and white in `color_vec3` (to match the third row of the tray).

7. Display the elements of each vector in the console (1st element, 2nd element, and 3rd element).

- The output should be:

```
Vector 1: blue white green
Vector 2: white green white
Vector 3: green blue white
```

8. Create a 2D vector (3 rows and 3 columns) of string object, named `color_vec_2d`

9. Use `color_vec1`, `color_vec2`, and `color_vec3` to build `color_vec_2d`

- First row of `color_vec_2d` consists of elements of `color_vec1`.
- Second row of `color_vec_2d` consists of elements of `color_vec2`.
- Third row of `color_vec_2d` consists of elements of `color_vec3`.

10. Display the size of `color_vec_2d`.

11. Read and display `color_vec_2d` in the console. The output should be:

```
blue  white  green
white  green  white
green  blue   white
```

Grading Rubric

5 pts– Comment your code (no need for Doxygen documentation):

- **0 pt**: No comment at all.
- **2.5 pts**: Partially commented.
- **5 pts**: Well commented.

5 pts– Variables naming based on the Google guidelines:

- **0 pt**: Did not follow the guidelines.
- **2.5 pts**: Partially followed the guidelines.
- **5 pts**: Followed the guidelines.

10 pts– Following assignment instructions:

- **2 pts**: Did not follow instructions.
- **5 pts**: Partially followed instructions.
- **10 pts**: Followed instructions.

5 pts– Output of your program is the same as step 11:

- **0 pt**: Output did not match at all step 11.

- **2.5 pts**: Output partially matched step 11.
- **5 pts**: Output perfectly matched step 11.

3 pts– Your code is clean, i.e., when looking at your code one may be able to clearly see the different steps of your program (increment your code, add comments, etc):

- **0 pt**: Code is a mess.
- **1.5 pts**: Code looks good but needs improvements.
- **3 pts**: Code looks good.

2 pts– Package and upload your project.

1. Name your project as follows: **RWA1-firstname-studentID**
2. Compress your project (e.g., **RWA1-firstname-studentID.zip**)
3. Upload your zipped project on Canvas.

- **0 pt**: Did not package project correctly.
- **2 pts**: Project packaged correctly.