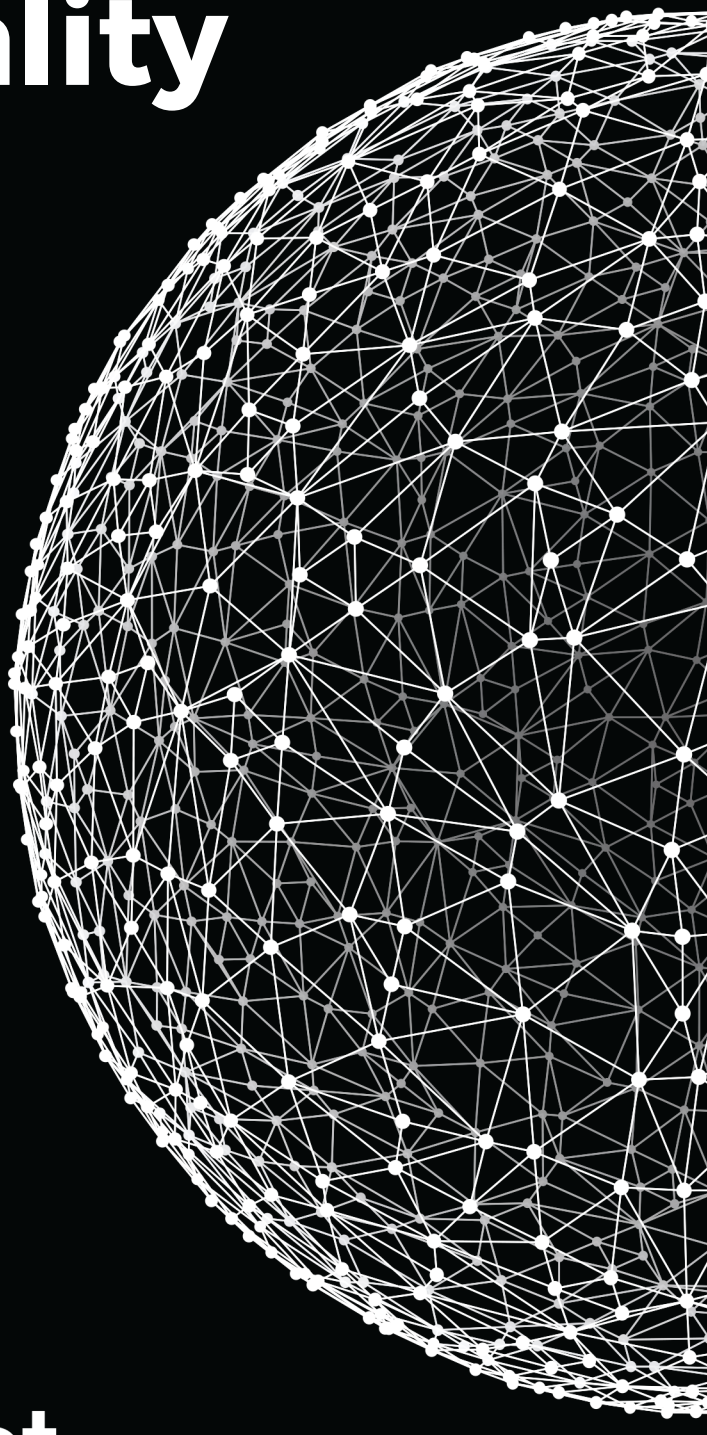
Race00: Virtual Reality

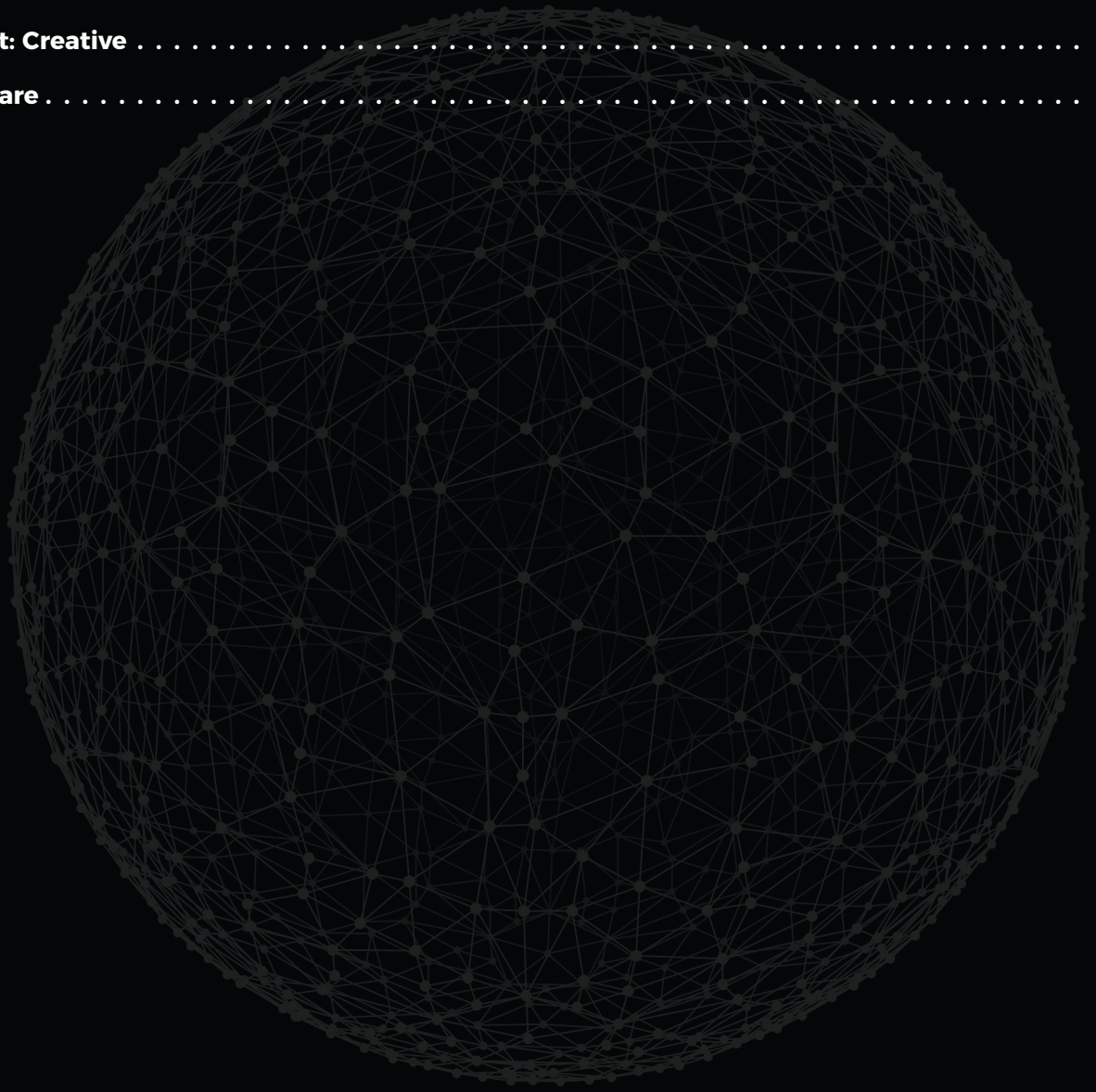
Marathon C

September 3, 2020



ucode connect

Contents



ucode connect

Engage

DESCRIPTION

Hi there!
Your first team challenge starts right now.

Teamwork skills are extremely important in this challenge. Overcoming the challenges of working in a team, you will gain experience that will be useful to you in any field. Communication, conflict management, listening, reliability, respectfulness are all about teamwork.

Teamwork isn't as easy as it seems. Be ready. Keep calm and work as a team.
In this challenge, you will learn how to create cubes and build pyramids, just like in Ancient Egypt. But instead of blocks, you will have ASCII characters.

Egyptian pyramids are big. How big will your cubes and pyramids be?

BIG IDEA

Teamwork.

ESSENTIAL QUESTION

How to build a powerful team?

CHALLENGE

Manage your teamwork to succeed with the solution.

Investigate

GUIDING QUESTIONS

We invite you to find answers to the following questions. By researching and answering them, you will gain the knowledge necessary to complete the challenge. To find answers, ask the students around you and search the internet. We encourage you to ask as many questions as possible. Note down your findings and discuss them with your peers.

- Have you worked in a team before? What was your teamwork experience?
- What makes a successful team?
- What team building activities do you know? Did you ever do any?
- What styles of teamwork do you know? What is a Teal Organisation?
- What skills and strengths are required to become an effective team member?
- What tools can help you improve your teamwork skills?
- How to use `git` when you write code in collaboration? What gitflow do you use?
- How to draw 3D objects on a 2D screen?
- What is the difference between a cube and a pyramid?

GUIDING ACTIVITIES

Complete the following activities. Don't forget that you have a limited time to overcome the challenge. Use it wisely. Distribute tasks correctly.

- Meet with your team. Discuss how you will organize teamwork. Create a channel for a team communication.
- Read the story, taking everyone's ideas on board.
- Have a cup of tea or coffee with your teammates and talk to each other about the challenge.
- Distribute tasks between all team members.
- Create a workplan for this challenge. Make sure the entire team understands what they need to do.
- Choose a gitflow for effective teamwork. We recommend you to use `git-flow-avh`.
- Clone your git repository that is issued on the challenge page in the LMS.
- Start to develop the solution. Offer improvements. Test your code.
- Communicate with students and share information.

ANALYSIS

Analyze your findings. What conclusions have you made after completing guiding questions and activities? In addition to your thoughts and conclusions, here are some more analysis results.

- Challenge has to be carried out by an entire team.

- Each team member must understand the challenge and realization, and be able to reproduce it individually.
- It is your responsibility to assemble the whole team. Phone calls, SMS, messengers are good ways to stay in touch.
- You can proceed to `Act: Creative` only after you have completed all requirements in `Act: Basic`. But before you begin to complete the challenge, pay attention to the program's architecture. Take into account the fact that many features indicated in the `Act: Creative` require special architecture. And in order not to rewrite all the code somewhen, we recommend you initially determine what exactly you will do in the future. Note that the `Act: Basic` part gives the minimum points to validate the challenge.
- Be attentive to all statements of the story.
- Analyze all information you have collected during the preparation stages. Try to define the order of your actions.
- Submit your files using the layout described in the story. Only useful files allowed, garbage shall not pass!
- Compile C-files with clang compiler and use these flags: `clang -std=c11 -Wall -Wextra -Werror -Wpedantic`.
- Pay attention to what is allowed. Use of forbidden stuff is considered a cheat and your challenge will be failed.
- Complete tasks according to the rules specified in the `Auditor`.
- The solution will be checked and graded by students like you. Peer-to-Peer learning.
- Also, the challenge will pass automatic evaluation which is called `Oracle`.
- If you have any questions or don't understand something, ask other students or just Google it.
- Use your brain and follow the white rabbit to prove that you are the Chosen one!

Act: Basic

DESCRIPTION

Do your best to create a positive atmosphere in your team. Speak up and be objective and, most importantly, effective. Your partner may have skills different from yours. Mutual assistance is always beneficial.

NAME

Virtual Reality: Cube

DIRECTORY

`./`

SUBMIT

`mx_cube.c, mx_printchar.c`

ALLOWED FUNCTIONS

`write`

LEGEND

M.: - I imagine you know something about virtual reality.
N.: - A little.
M.: - Tell me about it.
N.: - Essentially, it's a hardware system that uses an apparatus; headgear, gloves and whatever to make you feel that you are in a computer program.
M.: - If the virtual reality apparatus, as you called it, was wired to all of your senses and controlled them completely, would you be able to tell the difference between the virtual world and the real world?
N.: - You might not, no.
M.: - No, you wouldn't.

DESCRIPTION

Create a function `mx_cube` that prints a representation of a 3d cube to the standard ouput.
The size of the cube depends on the variable `n` which is passed to the function as a parameter.

There are some tips that help you to draw cube as it must be:

- you can use only `+`, `-`, `/`, `|` and `space` characters to draw a cube

- number of `|` in one vertical line equals to `n`
- number of `-` in one horizontal line equals to `2 * n`
- number of `/` in one diagonal line equals to `n / 2`

A cube can be produced only if `n > 1`. Otherwise, the program does not print anything.
This is it. Simple equations of the Matrix.

Oracle will compile your function as follows:

```
clang -std=c11 -Wall -Wextra -Werror -Wpedantic mx_cube.c main_cube.c mx_printchar.c -o mx_cube
```

See how the function must look like in the SYNOPSIS. Figure out how the program draws a cube in the CONSOLE OUTPUT section. Same style. Even the same characters.

SYNOPSIS

```
void mx_cube(int n);
```

CONSOLE OUTPUT

```
>./mx_cube | cat -e    # for n = 2
  +----+$
 /    /|$
+----+ |$
|    | +$
|    |/$
+----+$
>./mx_cube | cat -e    # for n = 3
  +------+$
 /      /|$
+------+ |$
|      | |$
|      | +$
|      |/$
+------+$
>./mx_cube | cat -e    # for n = 4
   +--------+$
  /        /|$
 /        / |$
+--------+  |$
|        |  |$
|        |  +$
|        | /$
|        |/$
+--------+$
>
```

CONSOLE OUTPUT

```
>./mx_cube | cat -e    # for n = 5
   +----------+$
  /          /|$
 /          / |$
+----------+  |$
|          |  |$
|          |  |$
|          |  +$
|          | /$
|          |/$
+----------+$
>./mx_cube | cat -e    # for n = 1
>
>./mx_cube | cat -e    # for n = 12
       +------------------------+$
      /                        /|$
     /                        / |$
    /                        /  |$
   /                        /   |$
  /                        /    |$
 /                        /     |$
+------------------------+      |$
|                        |      |$
|                        |      |$
|                        |      |$
|                        |      |$
|                        |      |$
|                        |      +$
|                        |     /$
|                        |    /$
|                        |   /$
|                        |  /$
|                        | /$
|                        |/$
+------------------------+$
>
```

Act: Creative

NAME

Virtual Reality: Pyramid

DIRECTORY

`./`

SUBMIT

`mx_pyramid.c, mx_printchar.c`

ALLOWED FUNCTIONS

`write`

LEGEND

– Something more. The Matrix is everywhere, it's all around us, here even in this room. You can see it out your window, or on your television. Even the Pyramids. You feel it when you go to work, or go to church or pay your taxes. It is the world that has been pulled over your eyes to blind you from the truth.

DESCRIPTION

Create a function `mx_pyramid` that prints a representation of a 3d pyramid to the standard output.
The size of the pyramid depends on the variable `n` which is passed to the function as a parameter.

There are some tips which help you to draw a pyramid as in the CONSOLE OUTPUT section:

- you can use only `/`, `\`, `|`, `_ (underscore)` and `space` to draw a pyramid
- the number of `/` in line that relates to the front triangle, equals to `n`
- the number of `\` in line that relates to the front and side triangles, equals to `n - 1`
- the number of `\` in line that relates only to the side triangle, equals to `n / 2`
- the number of `|` in vertical line equals to `n / 2`
- the number of `_` in horizontal line equals to `2 * n - 3`

A pyramid can be produced only if its size is even and `n > 1`. Otherwise, the program does not print anything.

Oracle will compile your function as follows:

```
clang -std=c11 -Wall -Wextra -Werror -Wpedantic mx_pyramid.c main_pyramid.c mx_printchar.c
-o mx_pyramid
```

See how the function must look like in the SYNOPSIS. Figure out how the program draws a pyramid in the CONSOLE OUTPUT section. Same style. Even the same characters.

SYNOPSIS

```
void mx_pyramid(int n);
```

CONSOLE OUTPUT

```
>./mx_pyramid | cat -e    # for n = 0
>./mx_pyramid | cat -e    # for n = 1
>./mx_pyramid | cat -e    # for n = 2
 /\$
/_\|$
>./mx_pyramid | cat -e    # for n = -2
>./mx_pyramid | cat -e    # for n = 4
   /\$
  / \ \$
 /   \ |$
/_____\|$
>./mx_pyramid | cat -e    # for n = 6
     /\$
    / \ \$
   /   \  \$
  /     \  |$
 /       \ |$
/_________\|$
>./mx_pyramid 7 | cat -e    # for n = 7
>
```

CONSOLE OUTPUT

```
>./mx_pyramid 42 | cat -e    # for n = 42
                                         /\$
                                        / \ \$
                                       /   \  \$
                                      /     \   \$
                                     /       \    \$
                                    /         \     \$
                                   /           \      \$
                                  /             \       \$
                                 /               \        \$
                                /                 \         \$
                               /                   \          \$
                              /                     \           \$
                             /                       \            \$
                            /                         \             \$
                           /                           \              \$
                          /                             \               \$
                         /                               \                \$
                        /                                 \                 \$
                       /                                   \                  \$
                      /                                     \                   \$
                     /                                       \                    \$
                    /                                         \                    |$
                   /                                           \                   |$
                  /                                             \                  |$
                 /                                               \                 |$
                /                                                 \                |$
               /                                                   \               |$
              /                                                     \              |$
             /                                                       \             |$
            /                                                         \            |$
           /                                                           \           |$
          /                                                             \          |$
         /                                                               \         |$
        /                                                                 \        |$
       /                                                                   \       |$
      /                                                                     \      |$
     /                                                                       \     |$
    /                                                                         \    |$
   /                                                                           \   |$
  /                                                                             \  |$
 /                                                                               \ |$
/_________________________________________________________________________________\|$
>
```

Share

PUBLISHING

Last but not least, the final stage of your work is to publish it. This allows you to share your challenges, solutions, and reflections with local and global audiences. During this stage, you will discover ways of getting external evaluation and feedback on your work. As a result, you will get the most out of the challenge, and get a better understanding of both your achievements and missteps.

To share your work, you can create:

- a text post, as a summary of your reflection
- charts, infographics or other ways to visualize your information
- a video, either of your work, or a reflection video
- an audio podcast. Record a story about your experience
- a photo report with a small post

Helpful tools:

- Canva - a good way to visualize your data
- QuickTime - an easy way to capture your screen, record video or audio

Examples of ways to share your experience:

- Facebook - create and share a post that will inspire your friends
- YouTube - upload an exciting video
- GitHub - share and describe your solution
- Telegraph - create a post that you can easily share on Telegram
- Instagram - share photos and stories from ucode. Don't forget to tag us :)

Share what you've learned and accomplished with your local community and the world. Use #ucode and #CBLWorld on social media.