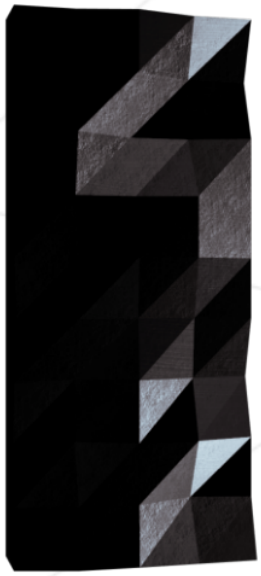

C Piscine

Rush 00

Summary: This document is the subject for Rush00 of the C Piscine @ 42.

Version: 5.2

Contents

Chapter I

Instructions

- The group WILL be registered to defense automatically.
- Do not cancel it, you won't get a second one.
- Any question concerning the subject would complicate the subject.
- You have to follow the submission procedures for all your exercises.
- This subject could change up to an hour before submission.
- The program must compile with the following flags: -Wall -Wextra -Werror; and uses `cc`.
- If your program doesn't compile, you'll get `0`.
- Your program must be written in accordance with the Norm. If you have bonus files/functions, they are included in the norm check and you will receive a 0 if there is a norm error inside.
- You will have to handle errors coherently. Feel free to either print an error message, or simply return control to the user.
- Rushes exercises have to be carried out by group of 2, 3 or 4.
- The mandatory rush number for your team will follow this rule:
  Alphabetical Index of the first letter of the team leader's login (from 1 to 26) modulo 5.
- You must therefore do the project with the imposed team and show up at Your defense slot, with all of your teammates.
- You project must be done by the time you get to defense. The purpose of defense is for you to present and explain your work.
- Each member of your group must be fully aware of the works of the project. Should you choose to split the workload, make sure you all understand what everybody's done. During the defense, you'll be asked questions and the final grade will be based on the worst explanations.

- It goes without saying, but gathering the group is your responsibility. You've got all the means to get in contact with your teammates: phone, email, carrier pigeon, spiritism, etc. So don't bother blurping up excuses. Life isn't fair, that's just the way it is.
- However, if you've really tried everything one of your teammates remains unreachable: do the project anyway, and we'll try and see what we can do about it during the defense. Even if the group leader is missing, you still have access to the submission directory.
- If you want bonus points, you may submit other subjects or be able to use program arguments to test your function.

Make sure the subject that was originally assigned to your group works perfectly before considering bonuses: If a bonus subject works, but the original one fails the tests, you'll get 0.

Chapter II

Foreword

Here are the lyrics of a famous TV show for everyone:

```
[Verse 1]
I wanna be the very best
Like no one ever was
To catch them is my real test
To train them is my cause

I will travel across the land
Searching far and wide
Each pokemon to understand
The power that's inside

[Chorus]
Pokemon! Gotta catch 'em all! It's you and me
I know it's my destiny,
Pokemon! Oh you're my best friend
In a world, we must defend
Pokemon! A heart so true
Our courage will pull us through,

You teach me and I'll teach you,
Pokemon! Gotta catch'em all

[Chorus]

Every challenge along the way
With courage I will face.
I will battle every day
To claim my rightful place.
Come with me,
The time is right,
There's no better team.
Arm in arm we'll win the fight!
It's always been our dream!
```

`[Chorus]`

I could bet you were singing right now, but it doesn't matter for the moment. And this subject is not related to Pocket Monster by the way...

Chapter III

Main subject

	Exercise 00
Rush0X	
Turn-in directory : *ex*00/	
Files to turn in : `main.c, ft_putchar.c, rush0X.c`	
Allowed functions : `write`	

- Files to submit: `main.c`, `ft_putchar.c` and your `rush0X.c`, '0X' represents the rush number. For example `rush00.c`.
- Those three files will be compiled together.
- Your file `ft_putchar.c` should include the function `ft_putchar`.
- Example of main.c:

```
int        main()
{
    rush(5, 5);
    return (0);
}
```

- You must therefore write the function `rush` taking two variables of type `int` as arguments, named respectively `x` and `y`. No need to say this function should be on the `rush0X.c` file.
- Your function `rush` should display (on-screen) a rectangle of `x` characters for width, and `y` characters for length.
- Your function should never crash or loop indefinitely.
- Your `main` will be modified during defense, to check if you've handled everything you're supposed to. Here's an example of test we'll perform:

```
int        main()
{
    rush(123, 42);
    return (0);
}
```

Chapter IV

Rush 00

- rush(5,3) should display:

```
$>./a.out
o---o
|   |
o---o
$>
```

- rush(5, 1) should display:

```
$>./a.out
o---o
$>
```

- rush(1, 1) should display:

```
$>./a.out
o
$>
```

- rush(1, 5) should display:

```
$>./a.out
o
|
|
|
o
$>
```

- rush(4, 4) should display:

```
$>./a.out
o--o
|  |
|  |
o--o
$>
```

Chapter V

Rush 01

- rush(5,3) should display :

```
$>./a.out
/***\
*   *
\***/
$>
```

- rush(5, 1) should display:

```
$>./a.out
/***\
$>
```

- rush(1, 1) should display:

```
$>./a.out
/
$>
```

- rush(1, 5) should display:

```
$>./a.out
/
*
*
*
\
$>
```

- rush(4, 4) should display:

```
$>./a.out
/**\
*  *
*  *
\**/
$>
```

Chapter VI

Rush 02

- rush(5,3) should display :

```
$>./a.out
ABBBA
B   B
CBBBC
$>
```

- rush(5, 1) should display:

```
$>./a.out
ABBBA
$>
```

- rush(1, 1) should display:

```
$>./a.out
A
$>
```

- rush(1, 5) should display:

```
$>./a.out
A
B
B
B
C
$>
```

- rush(4, 4) should display:

```
$>./a.out
ABBA
B  B
B  B
CBBC
$>
```

Chapter VII

Rush 03

- rush(5,3) should display :

```
$>./a.out
ABBBC
B   B
ABBBC
$>
```

- rush(5, 1) should display:

```
$>./a.out
ABBBC
$>
```

- rush(1, 1) should display:

```
$>./a.out
A
$>
```

- rush(1, 5) should display:

```
$>./a.out
A
B
B
B
A
$>
```

- rush(4, 4) should display:

```
$>./a.out
ABBC
B  B
B  B
ABBC
$>
```

Chapter VIII

Rush 04

- rush(5,3) should display :

```
$>./a.out
ABBBC
B   B
CBBBA
$>
```

- rush(5, 1) should display:

```
$>./a.out
ABBBC
$>
```

- rush(1, 1) should display:

```
$>./a.out
A
$>
```

- rush(1, 5) should display:

```
$>./a.out
A
B
B
B
C
$>
```

- rush(4, 4) should display:

```
$>./a.out
ABBC
B  B
B  B
CBBA
$>
```

Chapter IX

Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double check the names of your files to ensure they are correct.

As these assignments are not verified by a program, feel free to organize your files as you wish, as long as you turn in the mandatory files and comply with the requirements.

`You need to return only the files requested by the subject of this project.`