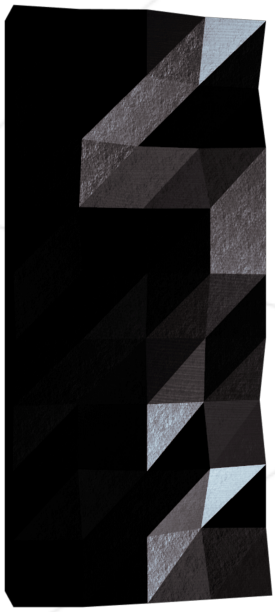

C Piscine

Rush 02

Staff 42 pedago@42.fr

Summary: This document is the subject for Rush02 of the C Piscine @ 42.

Contents

Chapter I

Instructions

- Each member of the group can register the whole group to defense.
- The group MUST be registered to defense.
- 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.
- These exercises are carefully laid out by order of difficulty - from easiest to hardest. We `will not` take into account a successfully completed harder exercise if an easier one is not perfectly functional.
- Moulinette compiles with the following flags: -Wall -Wextra -Werror; and uses `gcc`.
- If your program doesn't compile, you'll get `0`.
- Rushs exercises have to be carried out by group of 2, 3 or 4.
- In the group_promo.txt file, you'll find the list of imposed groups with the subject.
- You must therefore do the project with the imposed team and show up at the defense slot you've selected, 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 any and all details of 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 defense, you'll be asked questions, and the final grade will be based on the worst explainations.
- 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 always 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 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.

- It goes without saying, but your work must respect the Norm. Be thourough. And enjoy !

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 is a old-fashioned pecan pie recipe for you :

```
Ingredients
- Pastry dough
- 3/4 stick unsalted butter
- 1 1/4 cups packed light brown sugar
- 3/4 cup light corn syrup
- 2 teaspoon pure vanilla extract
- 1/2 teaspoon grated orange zest
- 1/4 teaspoon salt
- 3 large eggs
- 2 cups pecan halves (1/2 pound)
Accompaniment: whipped cream or vanilla ice cream

Preparation:
Preheat oven to 350°F with a baking sheet on middle rack.
Roll out dough on a lightly floured surface with a lightly floured rolling pin
into a 12 inch round and fit into a 9 inch pie plate.
Trim edge, leaving a 1/2-inch overhang.
Fold overhang under and lightly press against rim of pie plate, then crimp
decoratively.
Lightly prick bottom all over with a fork.
Chill until firm, at least 30 minutes (or freeze 10 minutes).
Meanwhile, melt butter in a small heavy saucepan over medium heat.
Add brown sugar, whisking until smooth.
Remove from heat and whisk in corn syrup, vanilla, zest, and salt.
Lightly beat eggs in a medium bowl, then whisk in corn syrup mixture.
Put pecans in pie shell and pour corn syrup mixture evenly over them.
Bake on hot baking sheet until filling is set, 50 minutes to 1 hour.
Cool completely.

Cooks notes:
Pie can be baked 1 day ahead and chilled. Bring to room temperature before
serving.
```

Chapter III

Subject

	Exercise 00
Rush-2	
Turn-in directory : $ex00/$	
Files to turn in : `Makefile and all necessary files`	
Allowed functions : `write, read, malloc, free`	
Notes : `n/a`	

- Create a program that takes a characters string as argument and displays the name of the rush in question, as well as its dimensions.
- Executable name: `colle-2`
- Your source code will be compiled as follows :

```
make fclean
make
```

- If the argument isn't a rush, here's an example of output display :

```
$> echo "Ai-je une chance d'avoir plus de 0 ?" | ./colle-2
aucune
$>
```

- Whatever the answer, your line must be ended by a "\n"
- If there is more than one rush, you must display them all alphabetically.

- Example :

```
$> ./colle-00 4 4
o--o
|  |
|  |
o--o
$> ./colle-00 4 4 | ./colle-2
[rush-00] [4] [4]
$> ./colle-01 3 4 | ./colle-2
[rush-01] [3] [4]
$> ./colle-02 1 1
A
$> ./colle-03 1 1
A
$> ./colle-04 1 1
A
$> ./colle-02 1 1 | ./colle-2
[colle-02] [1] [1] || [colle-03] [1] [1] || [colle-04] [1] [1]
$>
```

Chapter IV

Bonus

- Detection of any of those shapes is worth 2 points :

  1. rectangle
  2. square
  3. triangle
  4. lozenge

- This is on top of the mandatory subject.
- Example :

```
$> ./colle-02 3 3 | ./colle-2
[colle-02] [3] [3] || [carre] [3] [3] || [rectangle] [3] [3]
$>
```

- Make sure you have enough test files in order to support your work. You must demonstrate how your program detects shapes, and that it is indeed your `work`.

Chapter V

Super Bonus

This bonus doesn't require you having done the previous bonus, but the mandatory subject remains mandatory.

- Detection of any of those shapes is worth 2 points :

  1. sastantua
  2. reversed rectangle
  3. reversed square
  4. reversed triangle
  5. reversed lozenge
  6. reversed circle

- Make sure you have enough test files in order to support your work.
- A reversed shape is a shape made of spaces, the rest is made of characters.
- In this example, spaces are represented by dots ('.') :

```
$> cat ./carre_inverse_5_5
.....
.za0.
.za0.
.za0.
.....
$> cat ./carre_inverse_5_5 | ./colle-2
[carre inverse] [5] [5] || [carre] [3] [3]
```

- Any student who, on top of all these shapes, handles a reversed sastantua will get the ultimate grade of `42` for this project.