



C Piscine

Rush 02

Staff 42 [pedago@42.fr](mailto:pedago@42.fr)

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

# Contents

<b>I</b>	<b>Instructions</b>	<b>2</b>
<b>II</b>	<b>Foreword</b>	<b>4</b>
<b>III</b>	<b>Subject</b>	<b>5</b>
<b>IV</b>	<b>Bonus</b>	<b>7</b>

# 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.
- Your 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 explanations.
- 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 thorough. 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 : <i>ex00/</i>	
Files to turn in : <b>Makefile</b> and all necessary files	
Allowed functions : <b>write</b> , <b>malloc</b> , <b>free</b>	
Notes : <b>n/a</b>	

- Create a program that takes a number as argument and converts it to its written letters value.
- Executable name: **rush-02**
- Your source code will be compiled as follows :

```
make fclean
make
```

- If the argument isn't a valid integer, your program must return error, followed by a newline.
- Whatever the answer, your line must be ended by a newline.
- You can use the example executable to check your program.

- Example :

```
$> ./rush-02 42 | cat -e
forty two$
$> ./rush-02 0 | cat -e
zero$
$> ./rush-02 10.4 | cat -e
error$
$> ./rush-02 100000 | cat -e
one hundred thousand$
```

# Chapter IV

## Bonus

- Using `,` `,` `,` and `to` to be closer to the correct written syntax
- Doing the same exercise in a different language
- Using `read` to read standard entry when there is no argument
- any other bonus that you think bring something more to the project