



Codam Event!

The Not So Cold Race

Summary: This event is a small contest for fun!

Version:

Contents

I	Introduction	2
II	The task - Cold Race	3

Chapter I

Introduction

This contest is inspired by a rush that will soon come back into the Codam curriculum:

You need to produce a basic search engine and your code will be ranked in speed and accuracy with other contestants.

Every 3 hours after the beginning of the contest the actual ranking will be published.

The ranking will be based on what is on your Codam repository.

The contest will last for 24 hours.

You will ONLY be corrected by the ranking moulinette.

The contest only reward you with coalition points.

Chapter II

The task - Cold Race

Program name	coldrace
Turn in files	All your files
Makefile	all, clean, fclean, re, bonus
Arguments	
External functs.	<ul style="list-style-type: none">• open, close, read, write, printf, malloc, free, perror, strerror, exit• All functions of the math library (-lm man man 3 math)
Libft authorized	Yes
Description	Your program should read on the standard input to get pairs of key and value until an empty line is found. After the empty line the next values must be taken as a key request and the program should output the corresponding value is there's one or an error message.

The constraints are as follows:

- Name your program: **coldrace**.
- It takes no parameters.
- The first step is to store data as values associated to keys. The next one is to search them using keywords.
 - First, in storage mode the program reads on the standard input the keyword to store and, on the next line, its associated value until the first empty line.
 - After the first empty line the program switch to the search mode.
 - In search mode, it reads on the standard input the keyword to search and displays the result followed by a \n character on the standard output.

- If the keyword is not found, the program displays keyword: Not found. Replace “keyword” with the keyword that was searched.
- On the first empty line read on the standard input while in search mode, the program stops.
- the program should run as fast as possible.

Everyone with a functional program will receive a minimal amount of coalition points.

Because an example is often better than a thousand words, here is a sample of the input and output:

```
$>cat -n example
keyword$
value$
another-keyword$
another-value$
superultramegatestofcoldrace$
hey$
$
keyword$
another-value$
superultramegatestofcoldrace$
$
$>./coldrace < example
value
another-value: Not found.
hey
```



Note the line break that sends the data and separates them from the searches.