



# B1- C-Pool

---

B-CPE-042

## Day 12

---

File descriptors

v1.5



# Day 12

## File descriptors

---

binary name: : no binary  
repository name: : CPool\_Day12  
repository rights: : ramassage-tek  
language: : C  
group size: : 1  
compilation: : via Makefile, including *all*, *clean*, *fclean* and *re* rules

---



- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- All the bonus files (including a specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).



All **.c** files from your delivery folder will be collected and compiled with your **libmy**, which is found in **CPool\_Day12/lib/my**. For those of you using **.h** files, they must be located in **CPool\_Day12/include**. The Autograder will compile your library and your includes.



Allowed functions: open / read / write / close



Create your repository at the beginning of the day and submit your work on a regular basis!  
The delivery directory is specified within the instructions for each task.  
In order to keep your repository clean, pay attention to **gitignore**.



# Task 0

## Unit Tests

It is highly recommended to test your functions as you develop them. It is common practice to create a function named `main` (and a designated file to host it) to check the functions separately.

Create a directory named `tests`.

Create a `main` function within a file named `tests-$FUNCTION_NAME.c`, to be stored in the `tests` directory named. This function must contain all the necessary calls to the task function in order to cover all of the function's possible situations (normal or irregular).



The tests are not applicable for binaries

# Task 1

## Cat

Write a program called `cat`, which executes the same tasks as your system's `cat` command. You do not have to handle options.

There is an unlimited number of files given as parameter.

`cat` without parameters must be supported.

You must deliver a **Makefile** with the following rules: **all**, **clean**, **fclean** and **re**, and it must not relink. The binary's name must be `cat`.

You may use the `errno` variable (refer to `man errno`), but `perror()` and `malloc` functions are prohibited.

This task can only be performed by declaring a fixed-size array. It will have a limited size of approximately **30 ko**. In order to test the limitation, use the command `limit` in your shell:

```
Terminal
~/B-CPE-042> limit stacksize 32
~/B-CPE-042> limit stacksize
stacksize 32 kbytes
```



The read of size 1 is forbidden

**Delivery:** CPool\_Day12/cat/



Limit is an internal feature of a specific shell. Find the good one :)  
`man cat`



# Task 2

## Grep

Write a program called **grep**, which executes the same tasks as your system's **grep** command. You do not have to handle options.

There is an unlimited number of files given as parameter.

You must deliver a **Makefile** with the following rules: **all**, **clean**, **fclean** and **re**, and it must not relink. The binary's name must be **grep**.

You may use the **errno** variable (refer to **man errno**), but the **perror()** function is prohibited.

**Delivery:** CPool\_Day12/grep/



You don't have to handle regex. We are only asking for a simple matching system.  
`man grep`



The read of size 1 is forbidden

Terminal

```
~/B-CPE-042> ./grep looneytunes /etc/passwd
looneytunes:x:1000:100:looney tunes:/home/looneytunes:/bin/bash
```

Terminal

```
~/B-CPE-042> ./grep http /etc/services
http 80/tcp
...
```

Terminal

```
~/B-CPE-042> ./grep "application/pdf" /usr/share/misc/magic
!:mine application/pdf
```

Terminal

```
~/B-CPE-042> ./grep http /doesnt_exist
grep: /doesnt_exist: No such file or directory
```

Terminal

```
~/B-CPE-042> ./grep http /root
grep: /root: Permission denied
```