



B2- Elementary Programming in C

B-CPE-155

my_tar

Will you take some sauce with your steak?

v1.4



my_tar

Will you take some sauce with your steak?

binary name: my_archive, my_unarchive, my_tar
repository name: CPE_\$YEAR_b2rush1
repository rights: ramassage-tek
language: C
group size: 2
compilation: via Makefile, including re, clean and fclean 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 potential 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).

Authorized functions: all functions from the libC except functions that can be used to execute another binary (exec*, system, ...)



Your Makefile should build all the binaries by a simple call to 'make'



Except when it's explicitly mentioned, the grading will stop at the first task which is not fully fonctionnal



Step 0

You have to create three JPEG files:

- a hand drawn image of your favorite Epitech's Regional Pedagogical Director (DPR) called **dpr.jpg**
- a hand drawn image of your favorite Epitech's Regional's Development Director (DDR) called **ddr.jpg**
- a hand drawn image of your favorite Epitech's Educational Animator (AER/Astek) called **aer.jpg**



The files must be at the root of your directory.



Step 1

Write two programs: `my_archive` and `my_unarchive`.

`my_archive` should take one or more files in parameter and create an archive file that contains all of them.

Usage:

```
Terminal
~/B-CPE-155> ./my_archive
Usage: ./my_archive [archive_name] [files ...]
~/B-CPE-155> ls
file1 file2 file3 my_archive
~/B-CPE-155> ./my_archive toto file1 file2 file3
~/B-CPE-155> ls
file1 file2 file3 my_archive toto
```

`my_unarchive` should take an archive file in parameter and restore the files contained in it.

Usage:

```
Terminal
~/B-CPE-155> ./my_unarchive
Usage: ./my_unarchive [archive_name]
~/B-CPE-155> ls
my_unarchive toto
~/B-CPE-155> ./my_unarchive toto
~/B-CPE-155> ls
file1 file2 file3 my_unarchive toto
```



Step 2

To validate the second step, your binary `my_archive` have to archive directory recursively.
Your `my_unarchive` must be able to recreate the directories and files with the original directory tree.

```
Terminal
~/B-CPE-155> ls -R
.:
directory1 file1 file2 my_archive my_unarchive

./directory1:
directory2 file3 file4

./directory1/directory2:
file5
~/B-CPE-155> ./my_archive toto directory1 file1 file2
~/B-CPE-155> rm -r directory1 file1 file2
~/B-CPE-155> ./my_unarchive toto
~/B-CPE-155> ls -R
.:
directory1 file1 file2 my_archive my_unarchive toto

./directory1:
directory2 file3 file4

./directory1/directory2:
file5
```



Step 3

If fully functional, this step validate steps 1 and 2.

You have to make a program called `my_tar` that works as the unix `tar` and that is compliant with your programs `my_archive` and `my_unarchive` (it means that the format of your archives is up to you).

The arguments taken by your program have to be the same as the command `tar` of your system.

Options `xcft` are mandatory.

```
Terminal
~/B-CPE-155> ls -R
.:
directory1 file1 file2 my_archive my_tar my_unarchive

./directory1:
directory2 file3 file4

./directory1/directory2:
file5
~/B-CPE-155> ./my_tar cf toto file1 directory1 file2
~/B-CPE-155> rm -r directory1 file1 file2
~/B-CPE-155> ./my_tar xf toto
~/B-CPE-155> ls -R
.:
directory1 file1 file2 my_archive my_tar my_unarchive toto

./directory1:
directory2 file3 file4

./directory1/directory2:
file5
```



man tar



Step 4

Make your my_tar compliant with the tar format.

```
Terminal
~/B-CPE-155> ls -R
.:
directory1 file1 file2 my_archive my_tar my_unarchive

./directory1:
directory2 file3 file4

./directory1/directory2:
file5
~/B-CPE-155> ./my_tar cf toto.tar file1 directory1 file2
~/B-CPE-155> rm -r directory1 file1 file2
~/B-CPE-155> tar xf toto.tar
~/B-CPE-155> ls -R
.:
directory1 file1 file2 my_archive my_tar my_unarchive toto.tar

./directory1:
directory2 file3 file4

./directory1/directory2:
file5
```



Step 5

For this final step, you have to add the z option of compression to your my_tar. It has to be coded as the system tar and it has to compress on the fly.

For this step, you may use the zlib library which is already installed on your system. (link with -lz)



You must not have the archive compressed and uncompressed at the same time on your disk !

```
Terminal
~/B-CPE-155> ls -R
.:
directory1 file1 file2 my_archive my_tar my_unarchive

./directory1:
directory2 file3 file4

./directory1/directory2:
file5
~/B-CPE-155> ./my_tar czf toto.tar.gz file1 directory1 file2
~/B-CPE-155> rm -r directory1 file1 file2
~/B-CPE-155> ./my_tar xzf toto.tar.gz
~/B-CPE-155> ls -R
.:
directory1 file1 file2 my_archive my_tar my_unarchive toto.tar.gz

./directory1:
directory2 file3 file4

./directory1/directory2:
file5
~/B-CPE-155> rm -r directory1 file1 file2
~/B-CPE-155> tar xzf toto.tar.gz
~/B-CPE-155> ls -R
.:
directory1 file1 file2 my_archive my_tar my_unarchive toto.tar.gz

./directory1:
directory2 file3 file4

./directory1/directory2:
file5
```