

## B2 - Synthesis Pool

---

B-SYN-200

# automakefile

---

automake-like, but way better





# automakefile

binary name: skip-synthesis.sh, automakefile  
language: bash



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- 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).

## REQUIREMENT

Write a shell script named **skip-synthesis.sh** that will both read the output of the command `ls -l`, and print the user and the name of the files starting with the string given as parameter.

```
Terminal
~/B-SYN-200> ls -l | ./skip-synthesis.sh "to"
Zeus          totoTest
Zeus          tomatoKetchup
pierre.perret toutToutTout
Stallman      torus4d
```



The rest of the project will not be corrected unless this requirement is fully functional (and rewritten).



The file must be placed at the root of your git repository.



## MAKEFILE GENERATION

---

Write a shell script named **automakefile**, that generates a nice Makefile given a configuration file. This configuration file can contain the following (potentially unordered) lines:

- source\_filename;dependence1 dependence2...  
(specify the full names of the files, from the header subfolder below)
- PROJECT\_DIR;name\_of\_the\_project\_root\_folder
- SOURCES\_DIR;subfolder\_containing\_the\_source\_files
- HEADERS\_DIR;subfolder\_containing\_the\_header\_files
- LIBS\_DIR;subfolder\_containing\_libraries
- EXEC;executable\_name
- CC;compiler\_binary
- CFLAGS;compilation\_flag1 compilation\_flag1...
- LDFLAGS;linking\_flag1 linking\_flag2...



The generated Makefile must contain at least the classic clean, fclean and re rules.



PROJECT\_DIR is the only mandatory line, along with at least one source file.



## VERSION MANAGER

---

Add the following rules to the generated Makefile:

### **archive**

backup the source files in a compressed format, in the project folder

### **revert**

recreate the project from last backup

### **num**

print the current version number

### **delete**

delete last backup



While archiving, the message `xxx file generated.` must be printed, `xxx` being the name of the archive.

5 more fields will be necessary in your configuration file :

- BCK\_DIR;name\_of\_the\_backup\_folder
- ZIP;compression\_binary
- ZIPFLAGS;compression\_options
- UNZIP;decompression\_binary
- UNZIPFLAGS;decompression\_options



## EXAMPLE

```
Terminal
~/B-SYN-200> ls .
automakefile      fakeProj/
~/B-SYN-200> ls -R fakeProj/
fakeProj/:
calc.c
main.c
config

fakeProj/include:
calc/
fakeProj.h

fakeProj/include/calc:
calc.h

fakeProj/libs/:

fakeProj_backUp/:
```

```
Terminal
~/B-SYN-200> ./automakefile fakeProj/config && ls fakeProj/
calc.c
main.c
config
Makefile
```