

# GrayMeta Programming Test

Congratulations. You are a strong candidate and we would like to see what you can do.

This challenge contains a problem description and some requirements for the implementation. Please read the description and requirements carefully. Here's how your solution will be evaluated, in priority order (1 is highest priority):

1. Does the program have your name included in a comment?
2. Does the program compile and execute according to the problem description?
3. Does the program exhibit good design techniques?
4. Do you use data structures and algorithms that would allow your program to work under high volume – that is, is your code scalable?
5. Is your code well formatted and easy to read? Do you have sufficient comments? Can the reviewer quickly determine how your program is organized and what you are trying to accomplish by each class / method?
6. Have you thoroughly tested all possible input combinations?

Please write your code in Golang, Python, Perl, C/C++, PHP, BASH.

Thanks and good luck!

## Problem Description: Package Installation and System Dependencies

Dependencies between software packages are common in any Unix/Linux system. This means that to install a given software package, another package must be installed beforehand. Additionally lower level software components are often required by multiple upper level packages. For example, both the **telnet** client program and the **ftp** client program require the TCP/IP networking package. Conversely, the TCP/IP package cannot be removed until all packages that depend on it (telnet, ftp) are removed. We want you to design and write a program to automate the process of adding and removing software packages. To do this you will need to:

1. Maintain a record of installed packages and their dependencies.
2. Support explicitly installing a package in response to a command (unless it is already installed).
3. Support implicitly installing a package if it is needed to install another package.
4. Support explicitly removing a package in response to a command (if it is not needed to support other packages).
5. Support implicitly removing a package if it is no longer needed to support another component.

## Requirements

- Before installing a package, automatically install all the packages it requires.
- Before removing a package, confirm that no other packages require it. Dependent packages must be removed manually before the package can be removed.

## Input

The input received by your program will contain a sequence of commands, each on a separate line, containing no more than eighty characters. The command names (DEPEND, INSTALL, REMOVE, and LIST) always appear in uppercase starting in column one and the command line separator is one or more spaces. All appropriate DEPEND commands will appear before the occurrence of any INSTALL dependencies. The end of the input is marked by a line containing only the word END.

## Command Syntax

Syntax for commands is described in the table below:

| Command                    | Description  |
|----------------------------|--|
| DEPEND item1 item2 [item3] | Package item1 depends on package item2 (and item3 or any additional packages). |
| INSTALL item1              | Installs item1 and any other packages required by item1.                       |
| REMOVE item1               | Removes item1 and, if possible, packages required by item1.                    |
| LIST                       | Lists the names of all currently installed packages.                           |
| END                        | Marks the end of input, when used in a line by itself.                         |

## Output

1. Echo each line of input.
2. Follow each echoed INSTALL or REMOVE line with the actions taken in response, making certain that the actions are given in the proper order.
3. For the LIST command, display the names of the components currently installed.
4. For the DEPEND and END commands, no output, except the echo, is produced.
5. For the DEPEND command, there will only be one dependency list per item.

## Sample Input

```
DEPEND TELNET TCPIP NETCARD
DEPEND TCPIP NETCARD
DEPEND DNS TCPIP NETCARD
DEPEND BROWSER TCPIP HTML
INSTALL NETCARD
INSTALL TELNET
INSTALL foo
REMOVE NETCARD
INSTALL BROWSER
INSTALL DNS
LIST
REMOVE TELNET
REMOVE NETCARD
REMOVE DNS
REMOVE NETCARD
INSTALL NETCARD
REMOVE TCPIP
REMOVE BROWSER
REMOVE TCPIP
LIST
END
```

## Expected Sample Output

```
DEPEND TELNET TCPIP NETCARD
DEPEND TCPIP NETCARD
```

```
DEPEND DNS TCPIP NETCARD
DEPEND BROWSER TCPIP HTML
INSTALL NETCARD
NETCARD successfully installed
INSTALL TELNET
TCPIP successfully installed
TELNET successfully installed
INSTALL foo
foo successfully installed
REMOVE NETCARD
NETCARD is still needed.
INSTALL BROWSER
HTML successfully installed
BROWSER successfully installed
INSTALL DNS
DNS successfully installed
LIST
HTML
BROWSER
DNS
NETCARD
foo
TCPIP
TELNET
REMOVE TELNET
TELNET successfully removed
REMOVE NETCARD
NETCARD is still needed
REMOVE DNS
DNS successfully removed
REMOVE NETCARD
NETCARD is still needed
INSTALL NETCARD
NETCARD is already installed
REMOVE TCPIP
TCPIP is still needed
REMOVE BROWSER
BROWSER successfully removed
HTML is no longer needed
HTML successfully removed
TCPIP is no longer needed
TCPIP successfully removed
REMOVE TCPIP
TCPIP is not installed
LIST
NETCARD
foo
END
```

0.99.0 -