🡺 **Write a Pseudocode program independent of any computer language.**

All software languages have three kings of logic: sequential, decision (selection/conditional), and iterative (looping). Pseudocode has those same structural conventions but is intended for human reading rather than source code compiling. Pseudocode is a clear set of instructions, a procedure to do something, an algorithm.  
Pseudocode should be able to run on a human with pencil and paper.

Using the three kinds of logic, write the pseudocode instructions for a robot to…

* Select a dress shirt from the closet, put it on, and button it up.  
  *OR*
* Make a sandwich for you  
  hint: Google videos for “algorithm sandwich”

**N.B. How hard can it be?** There is much [tacit knowledge](https://en.wikipedia.org/wiki/Tacit_knowledge) you know *without thinking* that a robot also needs to complete the task without wrecking your home. E.g. you ensure a door is open before getting something out of a storage area.

Start

Go to the place where the closet is

Stand in front of the closet

Look for the closet’ handle

The robot tries to open the closet door

If the door is not open,

Find the key, then go to in front of the closet again.

Else, open the door

The robot checks what clothes are in the closet

If there is a no dress shirt in the closet

Close the closet and find another closet.

Else, there is a dress shirt, and then take it out.

The robot removes a hanger from the shirt and put it back to closet.

The robot holds and looks at the shirt.

If the buttons on the shirt are all locked,

Try to untie all the buttons.

Else, all the buttons are loose, grab the shoulder part of the shirt with the right hand.

The robot holds the right shoulder of the shirt

The robot puts its left hand on the left sleeve

The robot pulls the left shirt chest with its right hand

The robot makes the its left are all in the shirt

The robot bends its right arm as close to its body as possible

The robot holds the collar of the right shirt with its left hand

The robot puts its right arm as close to its body as possible

The robot bends its right elbow so that its right hand is in the belly button

The robot grabs the collar of the right shirt with its left hand

The robot lifts the collar over its shoulder

The robot puts its right hand in the right sleeve

When both arms properly dressed, pull the collar of the shirt once

The robot tries to lock one by one from the top button

If one button is misplaced

remove it again and put it back in the correct position

Else, button down to end of the bottom

The robot button up on the sleeve

End.

🡺 **Just how many different programming languages do we really need?**

One answer might be that there are different types of jobs a computer must do, and various languages are better suited to solving diverse kinds problems. OK, but how many languages and for what purpose?

On the other hand, as computer hardware gets faster and faster, the performance difference among languages and language types might become insignificant. Could a single language become the one universal all-purpose language?

Make a case in favour of the many and against the one (or the few).  
*OR*  
Make a case against the many and in favour of the one (or the few).

Programming languages have their own purposes and can solve problems by using the appropriate language. So I think that various languages are more suitable than one language. Computers do not really know what ‘programming language’ is, except ‘calculating and calculating which address the hardware should have a value of 0 or 1’.  
Programming is called conversation between people and computers. In other words, it is the process of talking to a computer by ordering it to perform a function. It is called dialogue, but in fact the order is 99%. The computer only responds to whether or not it carries out the command. It is the programming language that helps this process. The earliest computers were calculators made of iron and electricity. Therefore, there was no special need for dialogue. The purpose was to produce only the desired calculation value in the entered figure. So early programming was at the level of macro level for “calculation” rather than “language” tendencies. Then, as the value was compared and execution was needed, so that linguistic functions were needed. Eventually, people needed a language that created commands, put them in memory, and that was what we called “assembly”. This was considered essential until the eight-bit period when the computer itself was low. However, as computer hardware gradually developed and moved from 8 bit to 16 bit, humans needed a more comfortable language, and although various languages were developed and used, it was naturally a low-level language called “C” that contributed to human history. In addition to the memory computation, the C language was able to bring and use various branch doors, functions, and libraries, and it was very flexible. Of course, people still teach, learn, and use C language, but C language is too hardware-centric, so people described it as low-level. And the language developed for the purpose of human thought rather than hardware-oriented is described as advanced. People used basic language which is easier to understand and not much to do than C, and when there is a problem with hardware control and slowness of basic language, people used a high-level language called “pascal” because there are many restrictions. However, as the computer environment continues to grow faster and larger, it is OOP (Object Oriented Programming) that feels limited in software production in traditional ways. OOP is a method of programming with the “human world” mentality. The most popular language in this OOP was JAVA.

There are also many languages created for a particular purpose. Social networking services (SNS), which have become a daily routine for us today, are a web and internet- based world. There are also languages that specialize in web and Internet, such as Javascript, Python, and so on. Also, there are many platforms these days, and apps are produced according to those platforms. Such platforms include Windows, Mac, and Linux for desktops, and IOS(apple company devices) and Android for mobile devices. There are other platforms, but this is the most popular right now. In Windows, most programs use a high-level language called C++ and to develop IOS, their company introduced a new language called “Swift”, and Android mainly uses Java. As we already use a platform, we use different languages. It seems natural that as times develop, hardware develops, and new languages continue to emerge in line with that hardware. And because hardware is not just one language, it keeps developing in various languages. So it is impossible for a single language to replace all language and may rather prevent future development.  
However, there are about 600 programming languages now, but this seems to be too much. So it is not bad to combine as common languages as possible to relieve developers of their hardships.