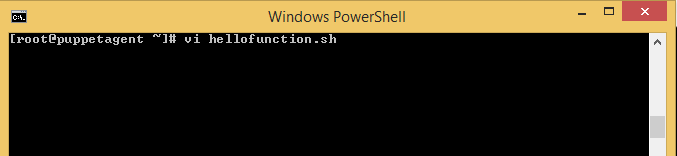
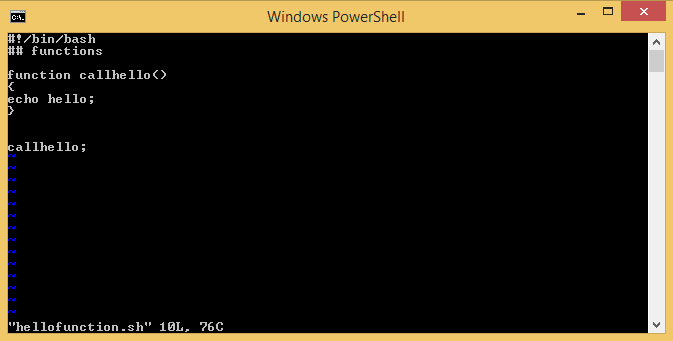
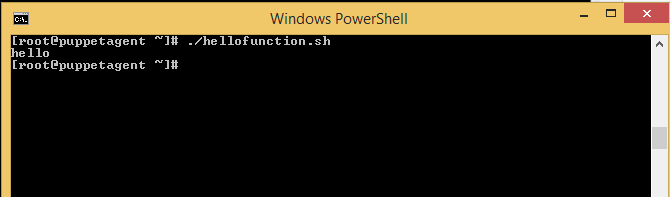
<https://www.youtube.com/watch?v=O4UJQA5sZEs&list=PL5nViEmyYI0bmfcM_s4P-dfzKrOpcFWWK&index=11>

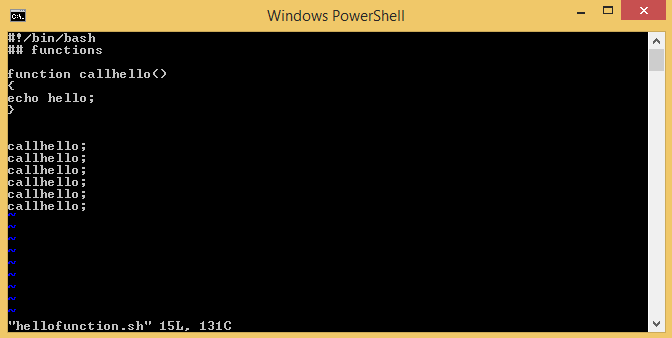


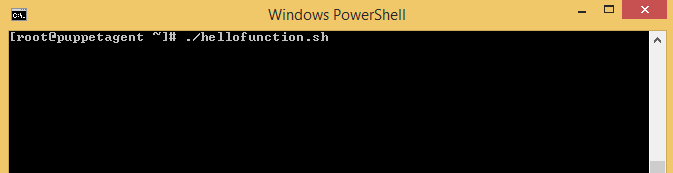


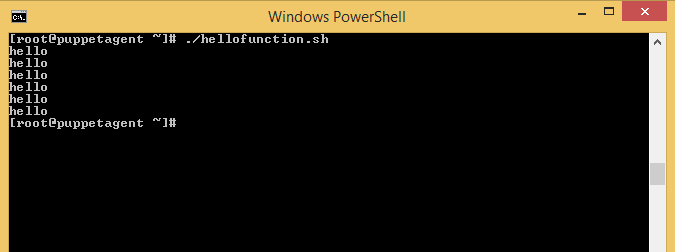




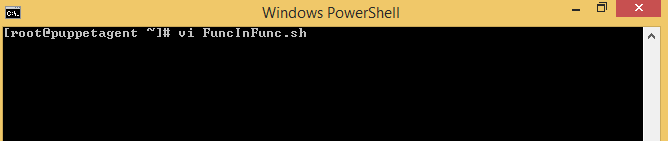


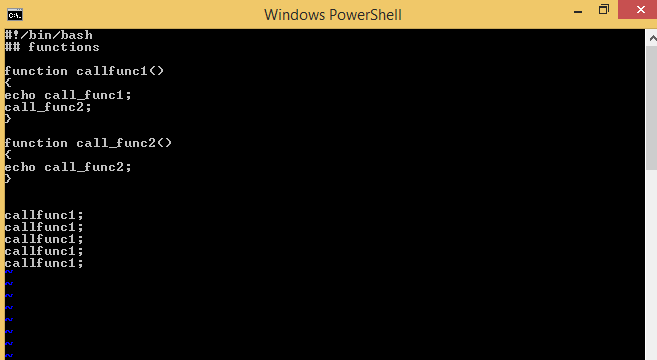


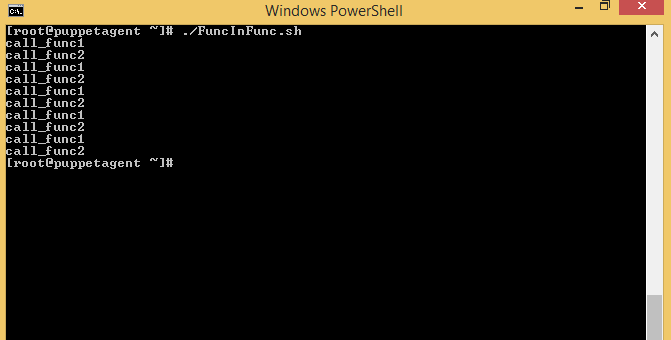




Call Function within Function





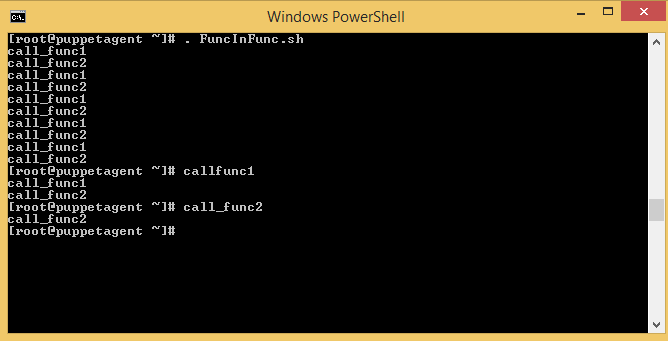


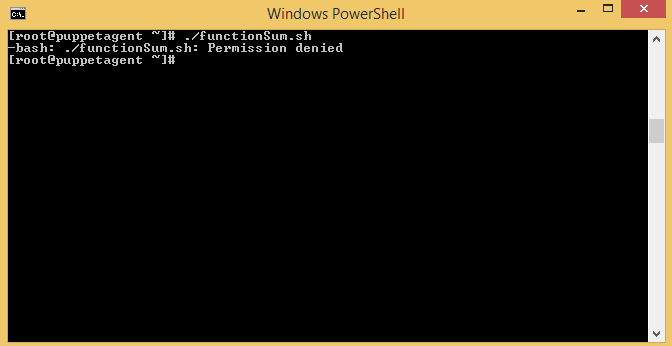
We can also call the function from command prompt.

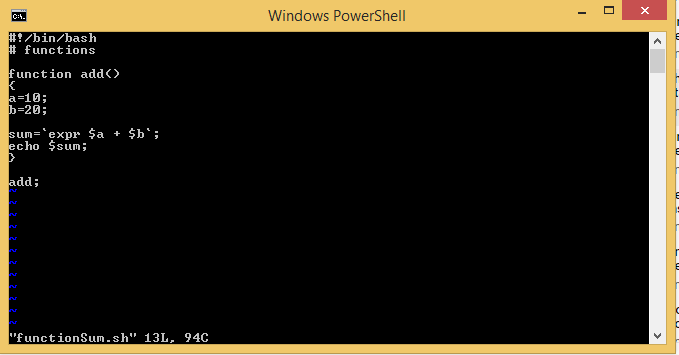
Type .(dot) function name.sh

. FuncInFunc.sh then

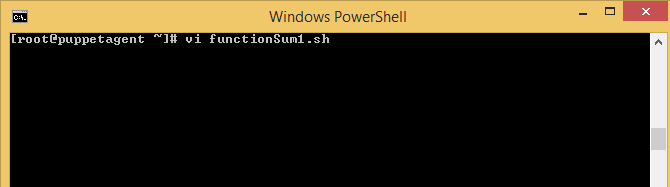
Call function name on command prompt

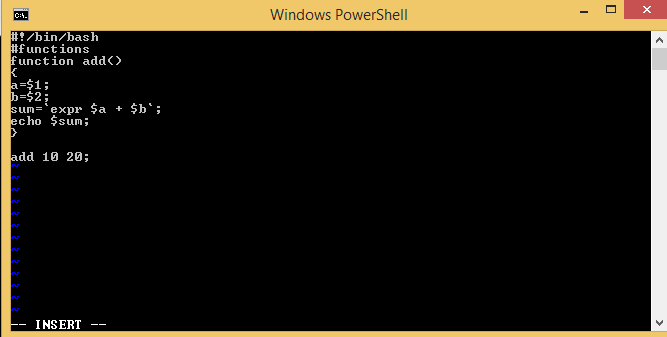


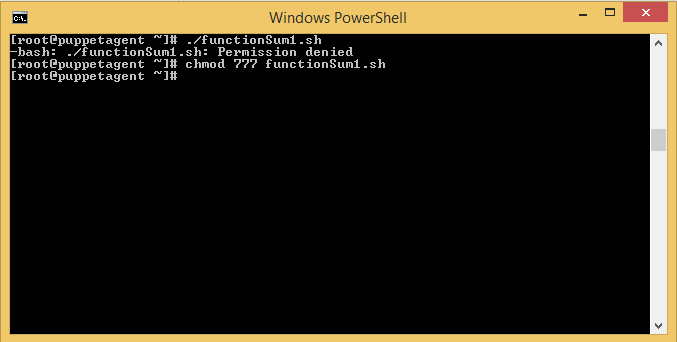


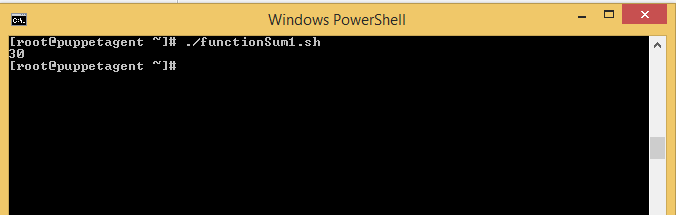


Passing parameter from command line arguments









**Copy, Cut and Paste in Normal Mode**

When you launch the Vim editor, you’re in the normal mode. In this mode, you can run Vim commands and navigate through the file.

To go back to normal mode from any other mode, just press the Esc key.

Vim has its own terminology for copying, cutting, and pasting. Copy is called yank (y), cut is called delete (d), and paste is called put (p).

**Copying (Yanking)**

To copy text, place the cursor in the desired location and press the y key followed by the movement command. Below are some helpful yanking commands:

* Yy-Yank (copy) the current line, including the newline character.
* 3yy - Yank (copy) three lines, starting from the line where the cursor is positioned.
* y$ - Yank (copy) everything from the cursor to the end of the line.
* y^ - Yank (copy) everything from the cursor to the start of the line.
* yw - Yank (copy) to the start of the next word.
* yiw – Yank (copy) the current word.
* y% - Yank (copy) to the matching character. By default supported pairs are (), {}, and []. Useful to copy text between matching brackets.

**Cutting (Deleting)**

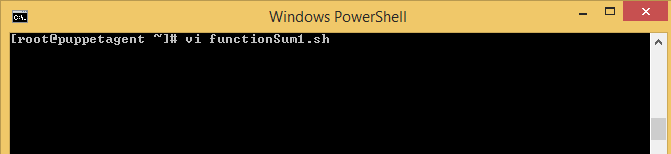
In normal mode, d is the key for cutting (deleting) text. Move the cursor to the desired position and press the d key, followed by the movement command. Here are some helpful deleting commands:

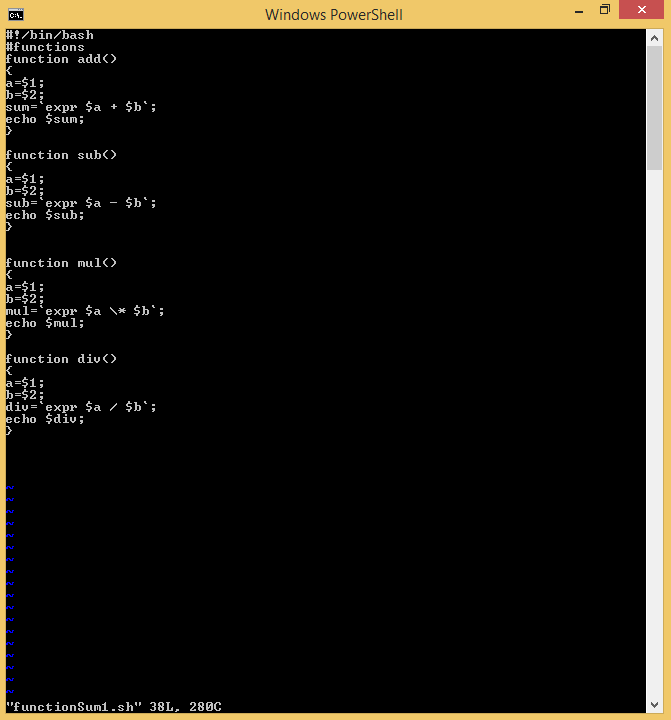
* dd - Delete (cut) the current line, including the newline character.
* 3dd - Delete (cut) three lines, starting from the line where the cursor is positioned,
* d$ - Delete (cut) everything from the cursor to the end of the line.

The movement commands that apply for yanking are also valid for deleting. For example dw, deletes to the start of the next word, and d^ deletes everything from the cursor to the start of the line.

**Pasting (Putting)**

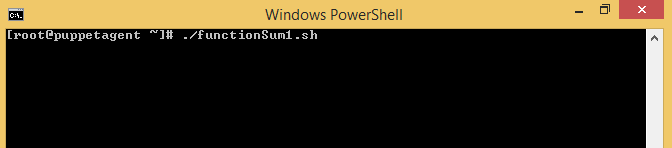
To put the yanked or deleted text, move the cursor to the desired location and press p to put (paste) the text after the cursor or P to put (paste) before the cursor.



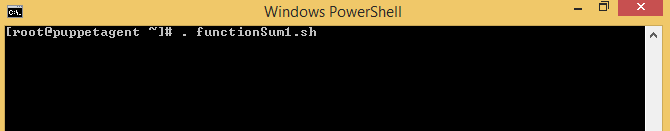


Follow the below step (compile) when you will make changes in program otherwise, otherwise changes will not reflect.

Step 1: Compile the program by ./functionSum1.sh



Called the program by .(dot) function name.sh(. functionSum1.sh) , so that we can call any function of the program from command prompt.



Now called every function of program by passing parameter as well.

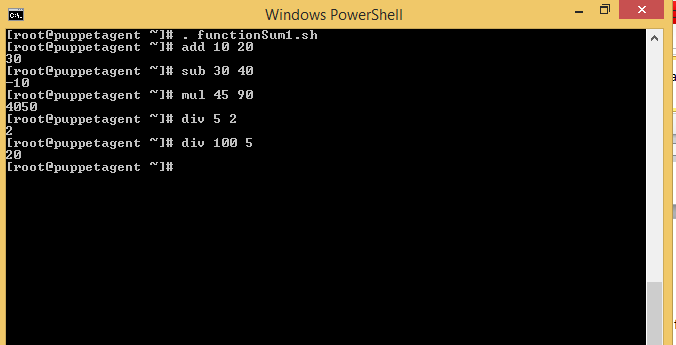
Add 10 20

Sub 30 40

Mul 45 90

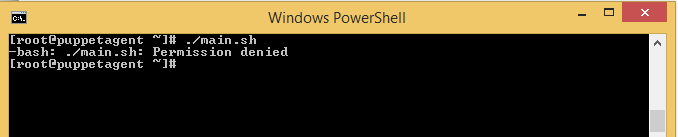
Div 5 2

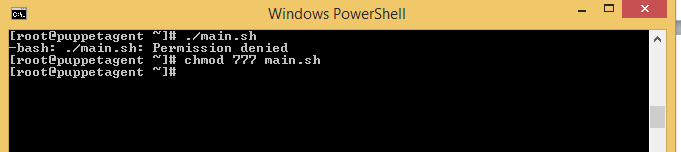
Div 100 5

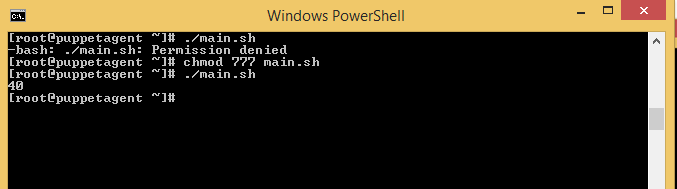


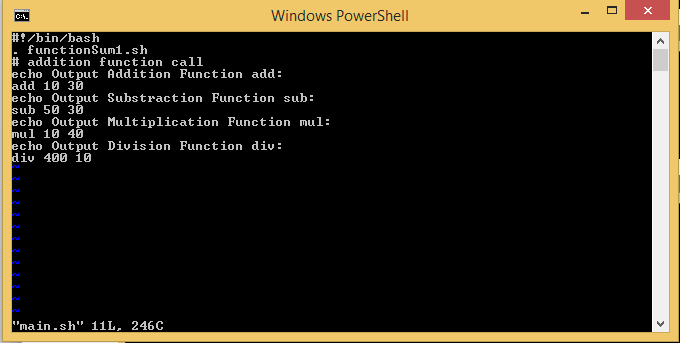
How to call function from other program(main.sh)

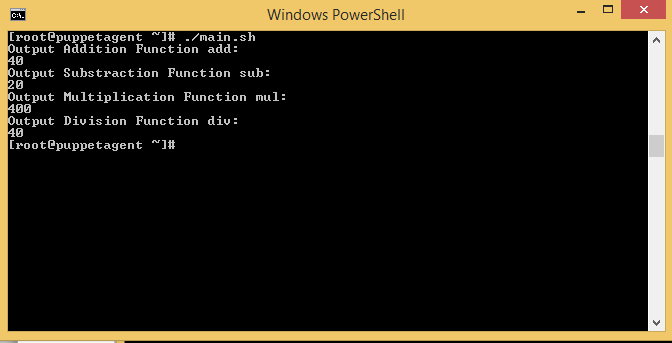


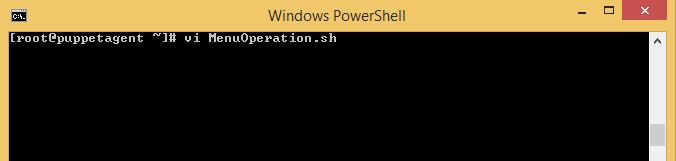




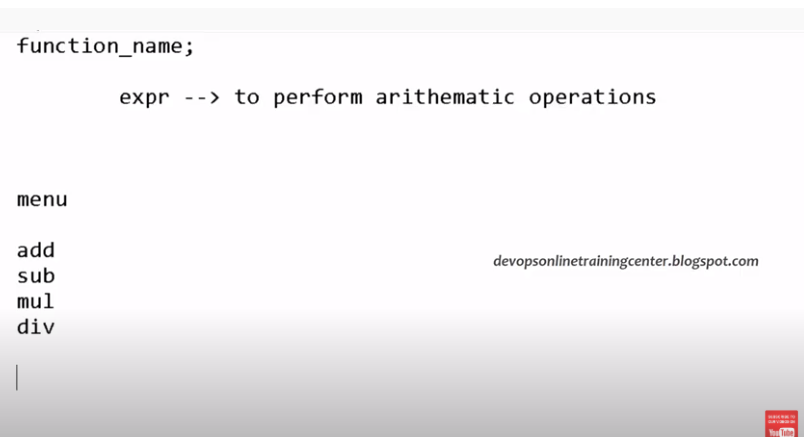


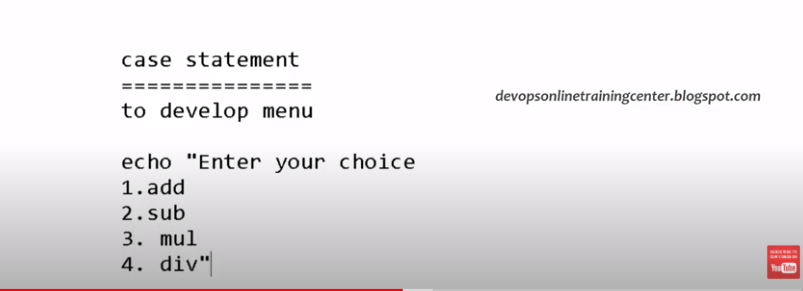




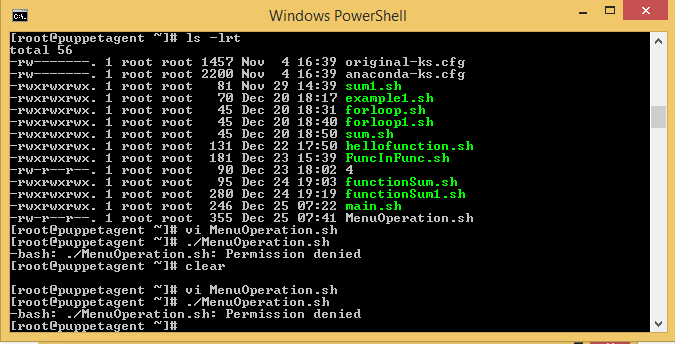


Create Menu option: Where User can select to do different Option like: add,sub,mul & Div

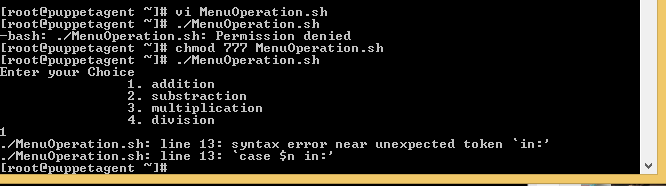




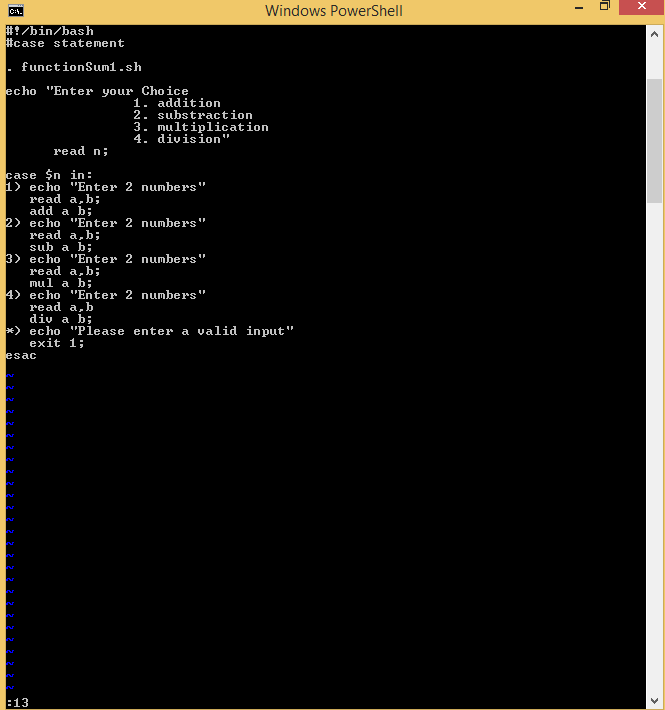
List out all the programs name in root path:



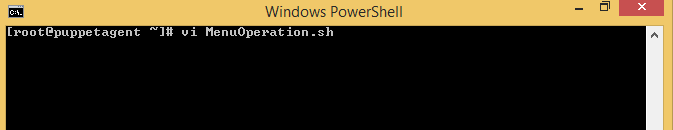
Error at line 13

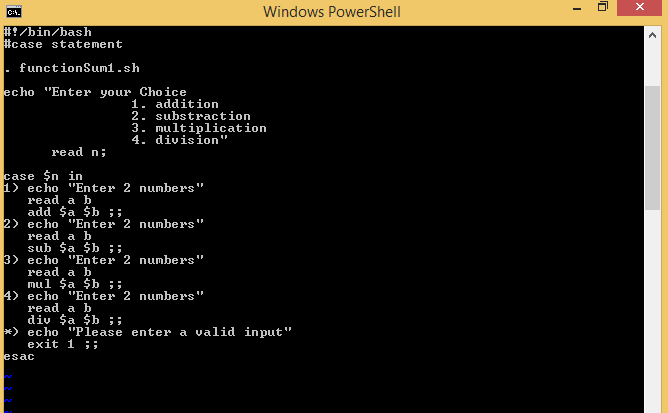


Now move cursor at line 13:

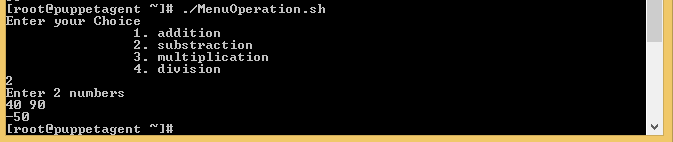


Correct Program below:

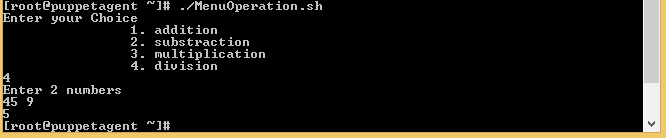




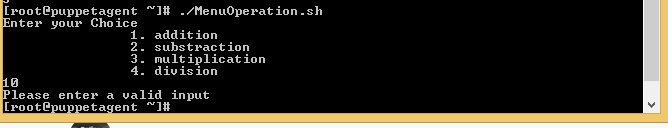


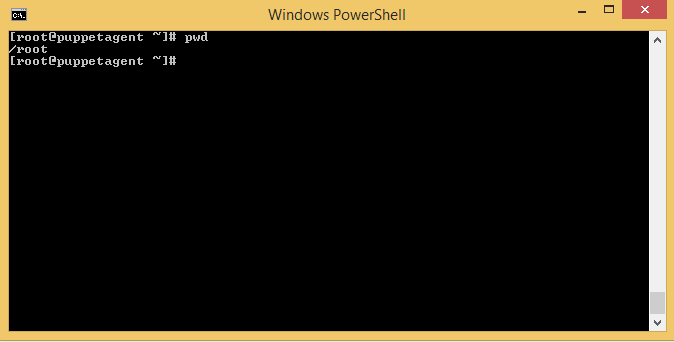




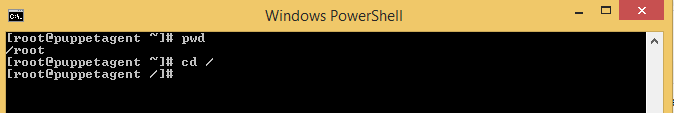


Enter Choice which unexpected:

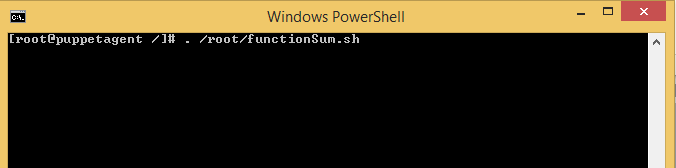
  
pwd: It is used to find the current working directory.



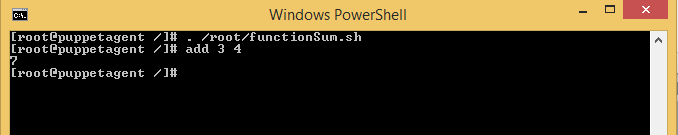
Change the path from root path by cd / command.



Export the program from root directory by following command:

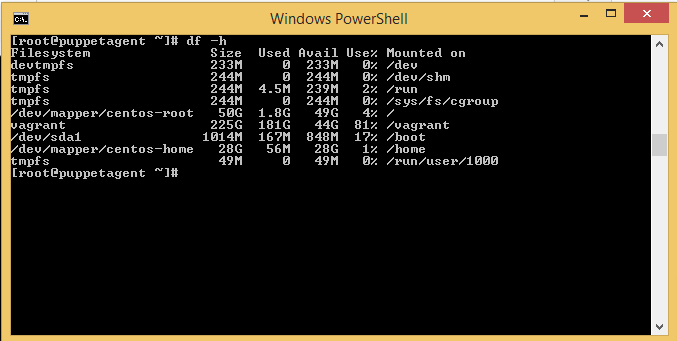


Now call the add method to add 2 numbers:





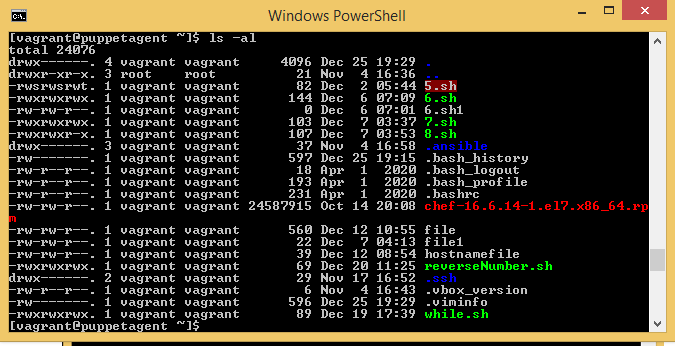
To see the system files for Vagrant user, which temporary



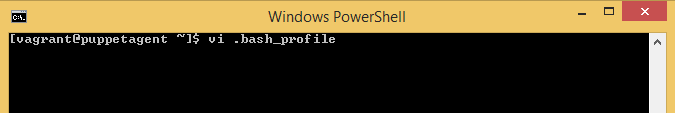
Suppose you like not to export function name after each time when login into vagrant.

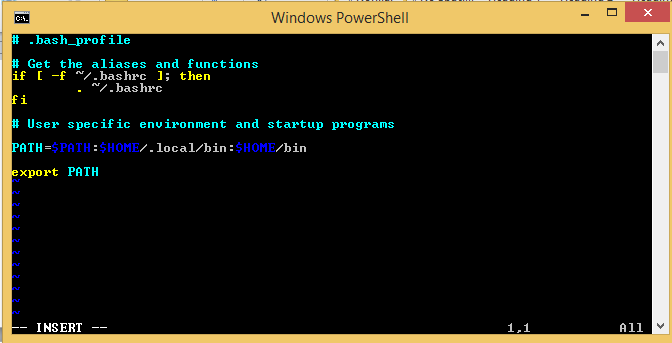
Function inside program should be called automatically. So to do that we need to write the program path into .bash\_profile file.

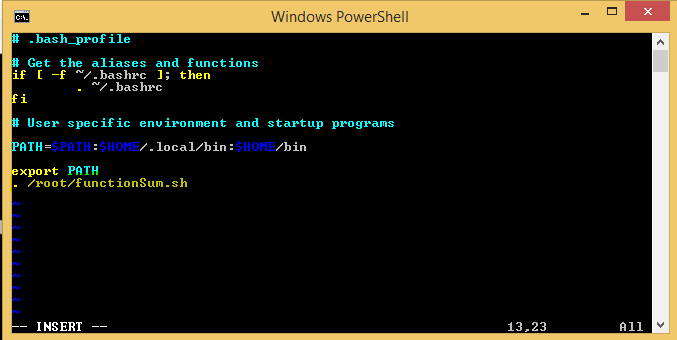
.bash files are hidden mostly. To see hidden files type “ls –al” command. We will get list of all hidden files.



Now open the .bash\_profile file and set the full path of your program. And program will be executed when user logged each time.All the functions are exported to command line.And that function will called from anywhere.

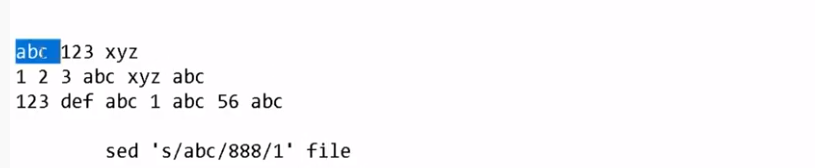








Unset command to clear the enviorment variables.



Sed: It is command used to replace the occurrence of character from lines.

Sed ‘s/abc/888/1’ file

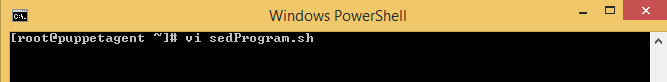
S: substitute

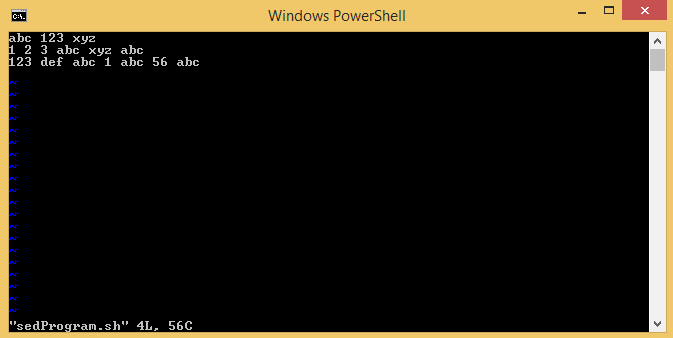
Abc:this character should be replaced

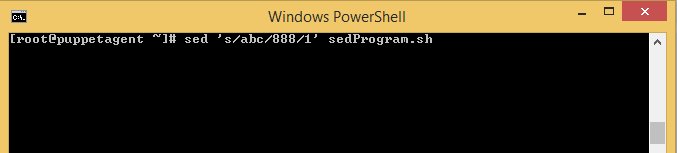
888:replace with triple 8

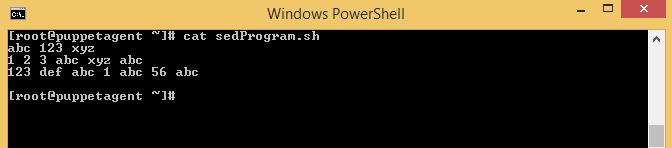
1: on first occurrence, It may be 2, 3,so on occurrence.

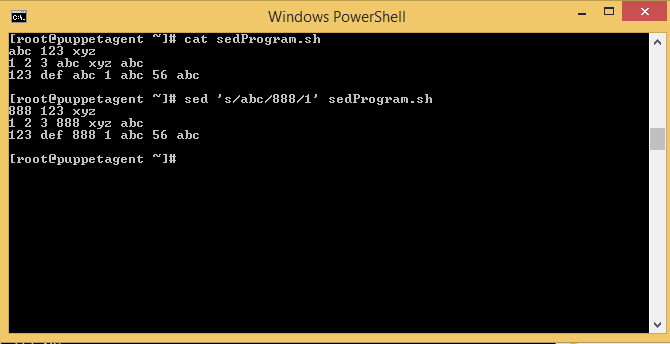
File:filename

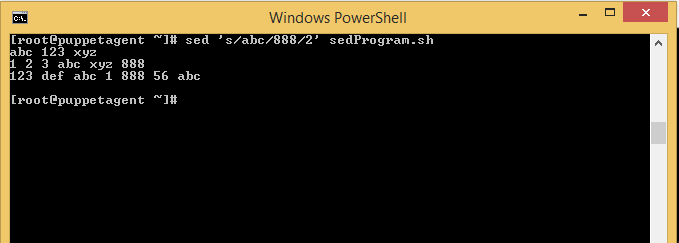










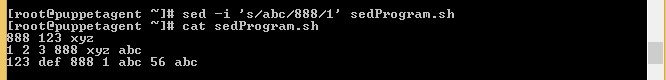




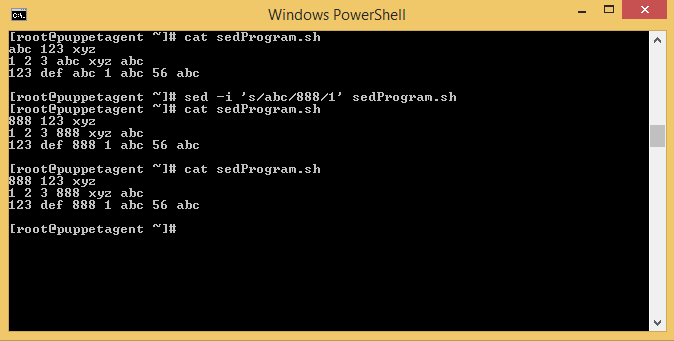
Sed is used for display purpose, they are not change permanent on file.

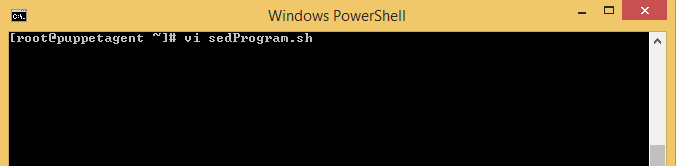
Permanent Change:

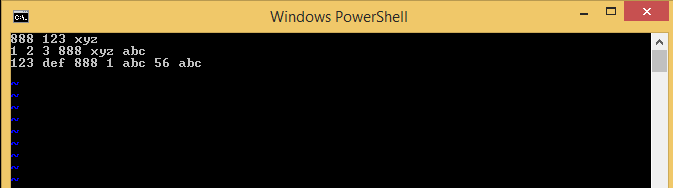






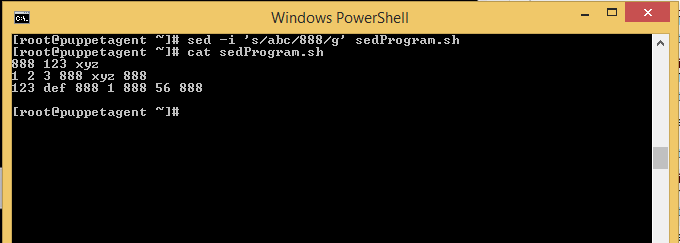






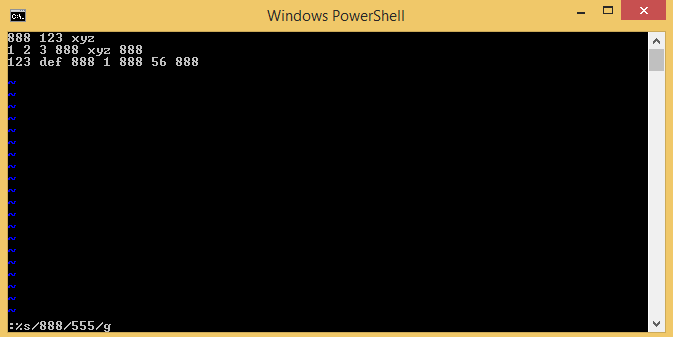
Sed –I ‘s/abc/888/g’ sedProgram.sh

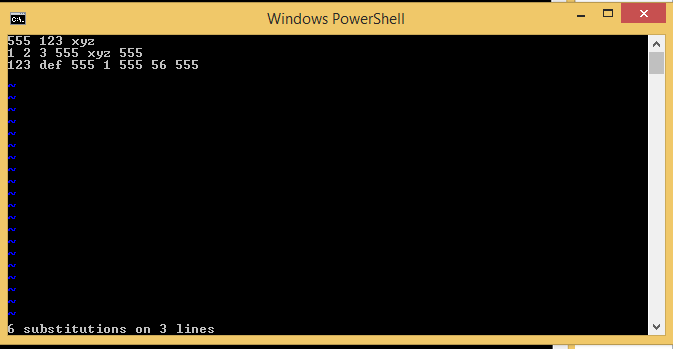
g-stands to change it globally, mean each occurrence of abc replace permanently with 888.



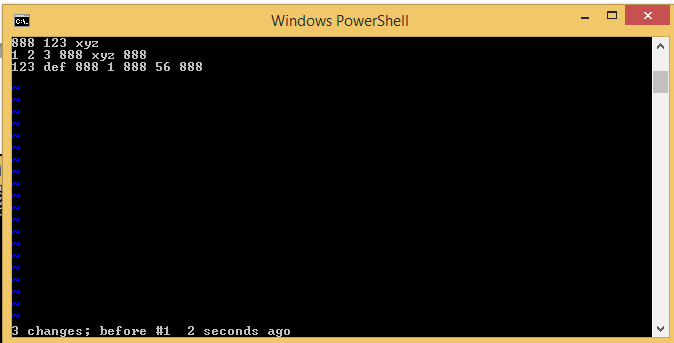
We can also change in the Vi editor. Replace the content in vi editor.

With Esc+U we can come back to orginal content in vi editor, but if we change it on command line, then it become permanent change, not reverseable.

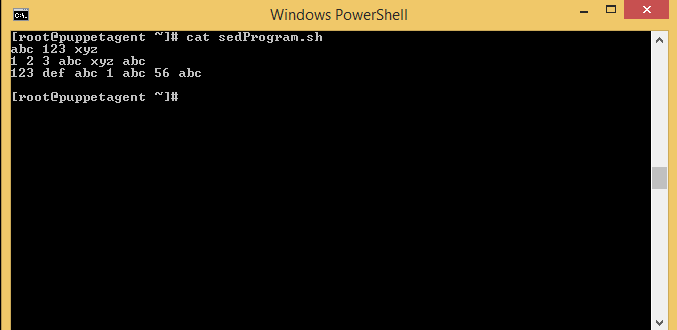


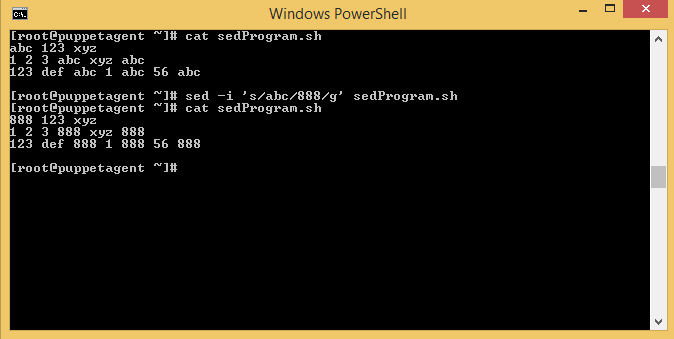


With help of ESC+U, we can undo the changes, back to the content in Original state.



But changes on Command line is permanent.







Sed: stands for FIND and REPLACE