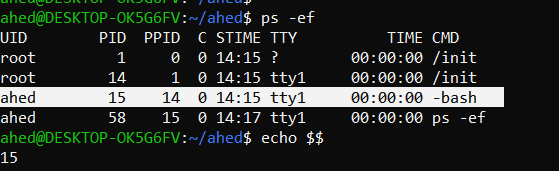
Shell script part 2

1- The $$ variable:

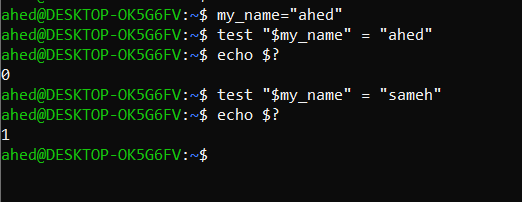
* يحتوي على قيمة pid للبروسيس process الي شغالة حاليا :



* تستخدم مع kill اذا بدي انهي ال procees الشغالة
* وتستخدم ايضا مع multiprocessing يعني اذا كل بروسيس بدها تنشأ فايل خاص فيها منقدر نميز كل فايل ب pid الي بقدر اجيبه من خلال $$

2- test command:

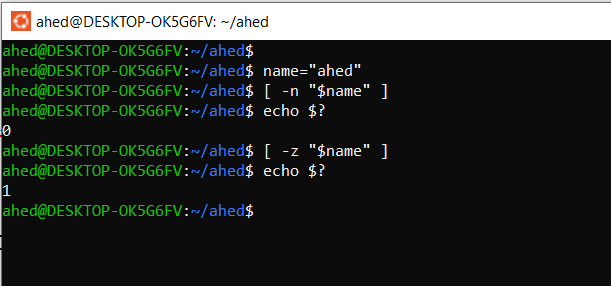
* يستخدم لفحص قيمة او اي شرط
* كيف بستخدمها ؟ كالتالي test expression او [ expression ]
* **test string :**



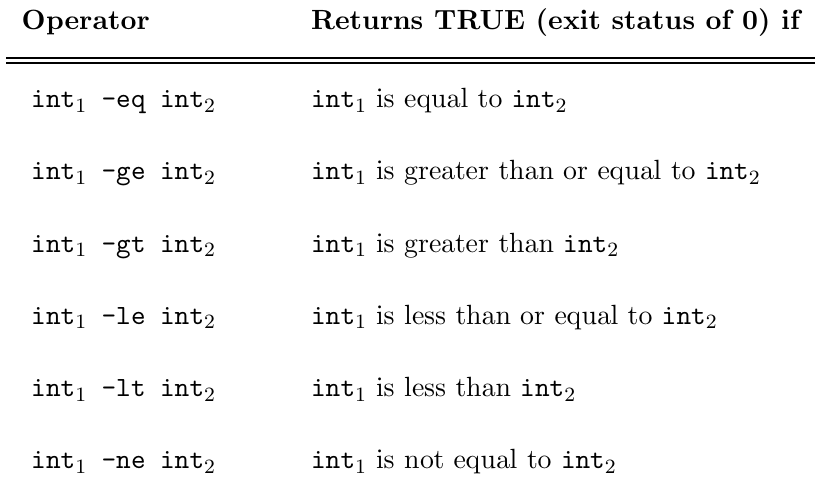
* في هذا المثال حطينا في المتغير my\_name قيمة معينة "ahed" وقارنت قيمة هاد المتغير ب ahed من خلال test، في هاي الحالة القيمة الي في my\_name هي نفس القيمة الي قارنت فيها ف لازم يعطيني true
* عشان اعرف النتيجة لازم اطبع exit status الي ?$ وكما هو واضح طبعت 0 يعني true
* وفي المقارنة الثانية قارن قيمة المتغير "ahed" ب قيمة "sameh" وطبعا هدول مش متساويات فلازم يعطيني false
* كمان مرة بطبع exit status و طبعت 1 يعني false (اي رقم موجب في exit status يمثل false )
* مهممممم : لازم احط space قبل وبعد =
* مهممممم: الافضل نحط double quotation على الاشياء الي بقارنها الي هم قبل وبعد = ، مثلا حسب المثال حطينا my\_name و القيمة الي بعد = في "" —-> "my\_name$"

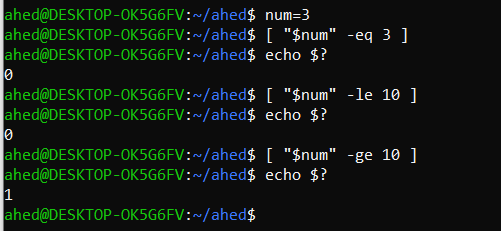
* String operation :

| operation | value |
| --- | --- |
| string1 = string2 | 0 if two strings are equal, else return 1 |
| string1 != string2 | 0 if two strings are not equal, else return 1 |
| -n string | 0 if string is not empty, else return 1 |
| -z string | 0 if string is empty, else return 1 |

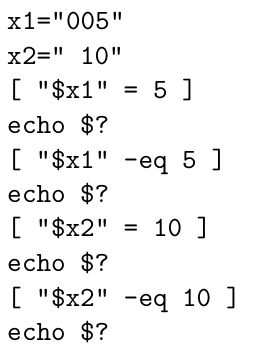


* **Test integers :**
* تستخدم لعمل مقارانات على الارقام الصحيحة

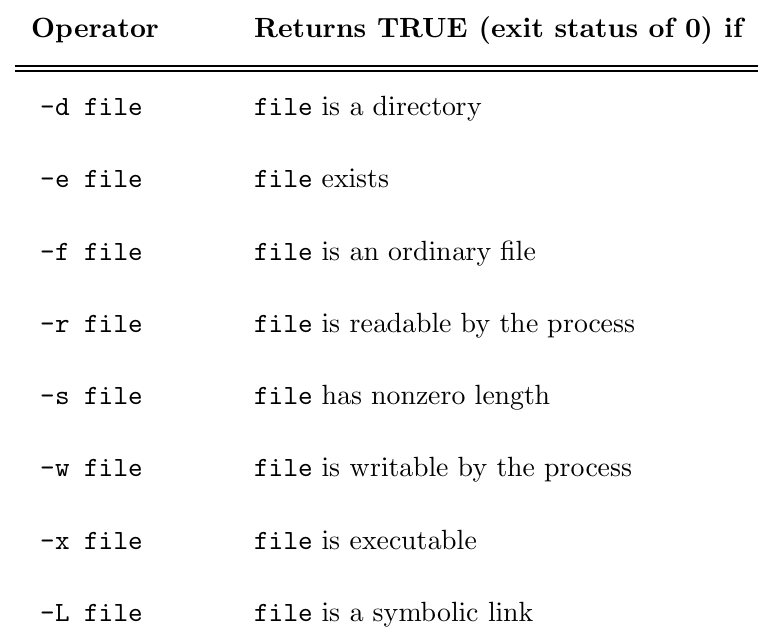


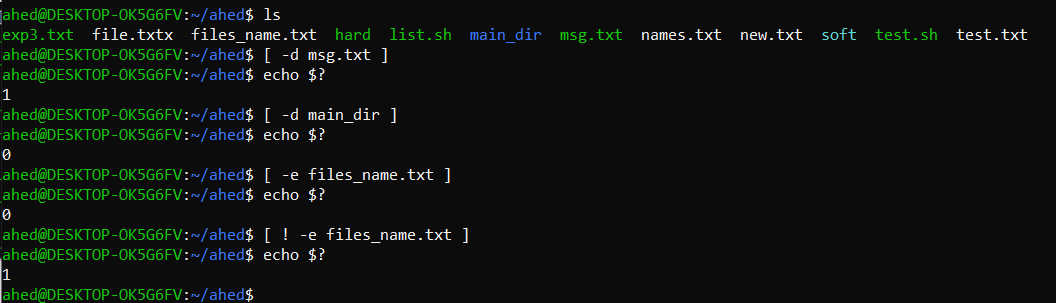


Q) what is the output of the following:



* **test files:**
* تستخدم لفحص الملفات مثلا هل هذا الملف ملف عادي وانه مش مجلد او هل هذا الملف موجود او هل يملك صلاحية القراءة او هل حجمه صفر ….. الخ.

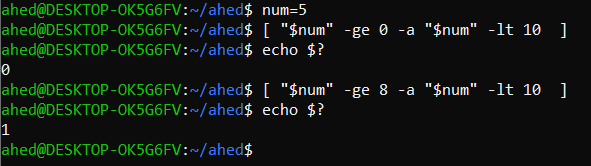




note) ! means not

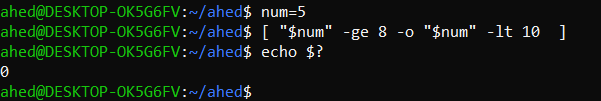
3- The Logical AND Operator -a:

* منستخدمها لما بدنا نفحص اكثر من شرط واحد
* مثلا بدنا نفصحص قيمة رقم معين هل هو بين 0 و 10 يعني اكبر من 0 واقل من 10 :

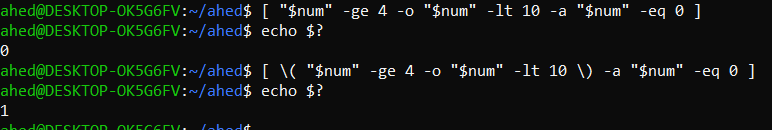


* Q) What is another way to do the same thing????

4- The Logical OR Operator -o:



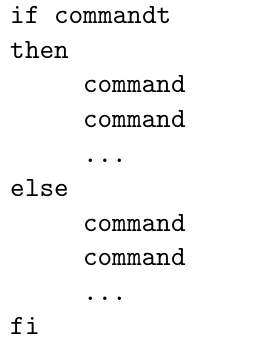
* اذا استخدمنا a- و o- مع بعض تكون الاولوية ل a-
* اذا اردنا تغير الاولوية نستخد (\ )\ :



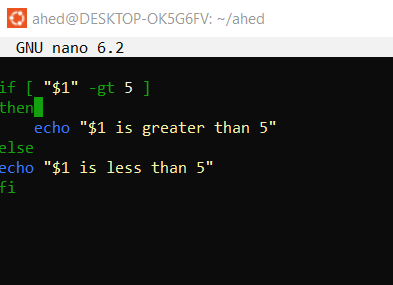
Q) explain the output (why in the first case the output is 0 and in the second case the output is 1)?

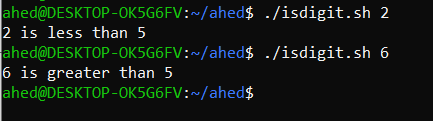
5- if else condition:

* تستخدم لعمل شيء معين اذا تحقق شرط معين
* Syntax :

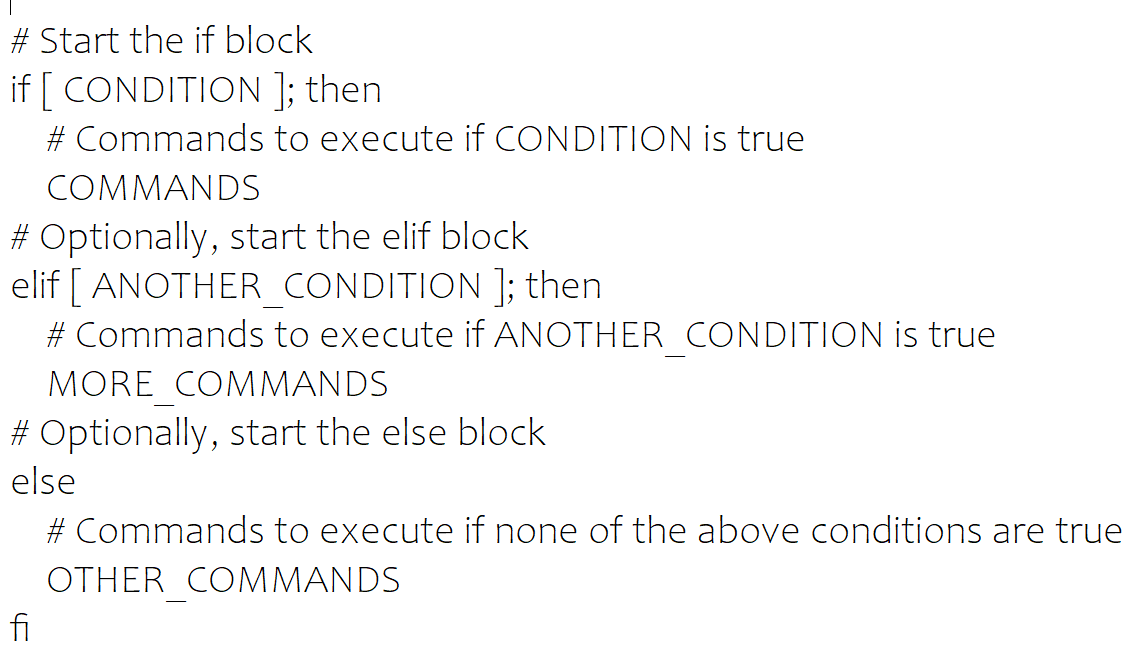


* مهمممممممم: ما تنسى fi ، اذا ما حطيتها رح يعطيك ايرور EOF
* نشوف مثال، خلينا نكنتب shell script بتوخذ رقم ك argument وبتفحص هل الرقم اكبر من 5 :



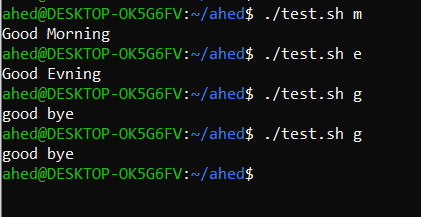


6- if elif :



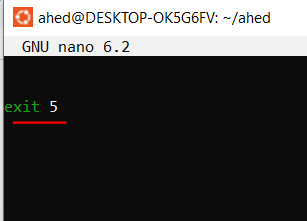
* نفس if else if في لغة السي
* خلينا نكتب shell script بتوخذ argument ، اذا كانت arg هي حرف m رح يطبع good morning
* اذا كانت arg هي حرف e رح يطبع good evning
* اي شي ثاني يطبع good bye



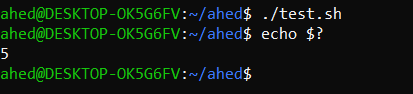


7- exit n : n is any integer number

* تستخدم للخروج من التيرمنال او من shell script ويمكن تحديد رقم معها ويكون ك exit status (?$)



هسا لو اطبع exit status بعد تشغيل shell script رح يطبعلي 5



Q1) write a shell script that takes two numbers as arguments, and check if there are two arguments then print the sum of them else exit from the program.

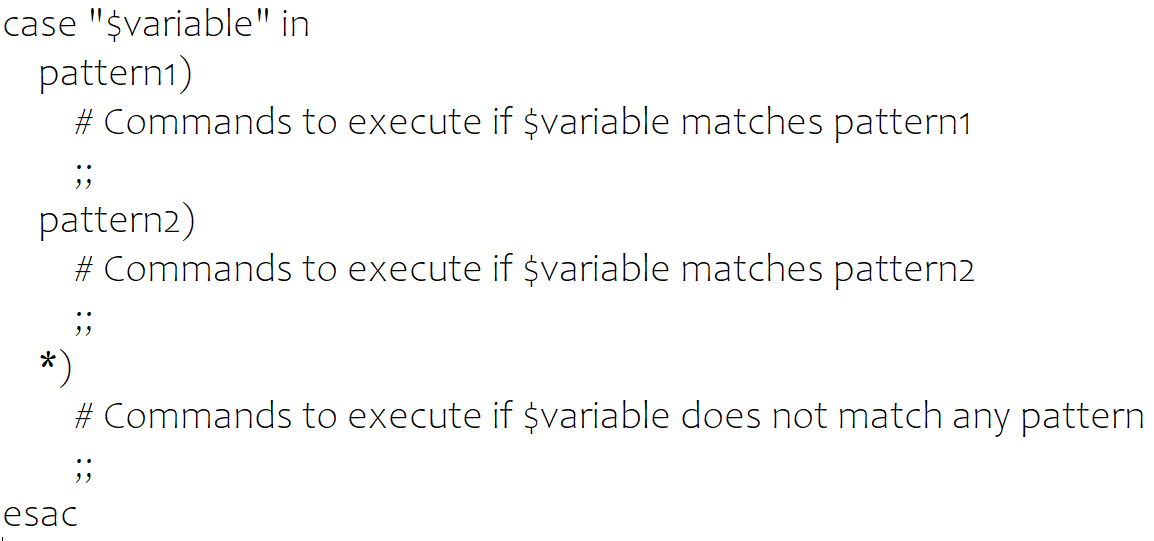
Q2) write a shell script that takes a string as an argument and print its length. Also check if there is only one argument.

Q3) write a shell script that prints “good morning” if the current hour is less than 12 pm, otherwise print “good evening”.

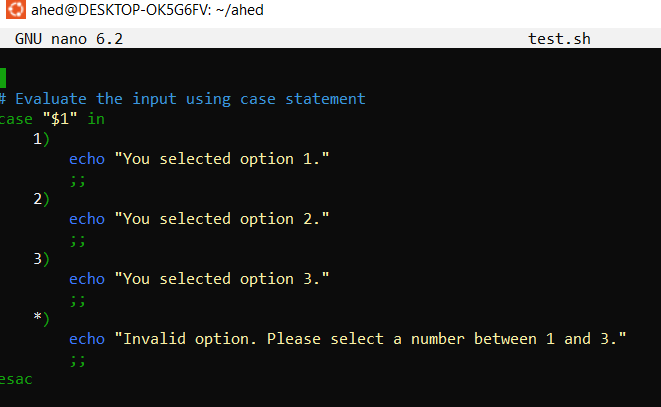
Q4) write a shell script that check if the specific user is login to the system or not, you must pass the name of the user as argument.

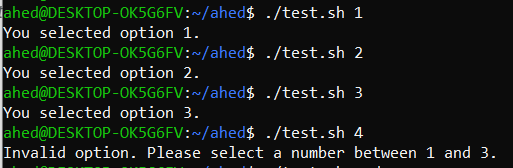
8- case (switch):

* Syntax :

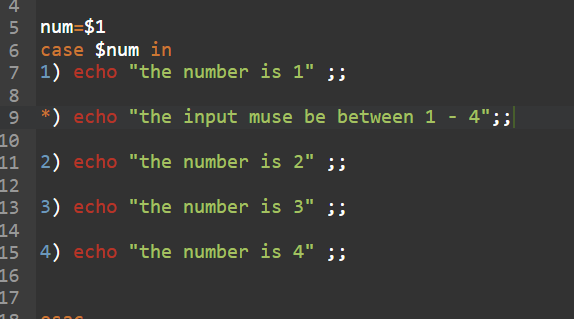


* مهمممممم : ما تنسى esac
* مثال: خلينا نكتب كود بفحص اذا الرقم المدخل ك arg هو بين 1 و 3 واذا دخل اي شي ثاني يطبعله انه لازم يدخل من 1 ل 3:





* مهممممممممم: ما تحطو defualt case ( (\* ) اول شي او ثاني ، لازم تكون اخر case عشان يعطيكم حل صح

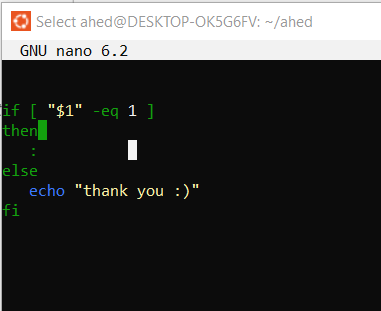


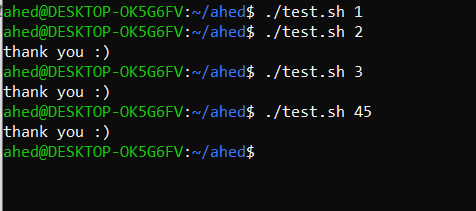
* زي الكود الي مكتوب بالصورة غلط يكون هيك لانه رح يعطينا logic غلط

Q) write a shell script that takes a character as an argument, then checks if the character is a digit or small letter or capital letter, otherwise it will print a special character.

9- The Null Command (:) :

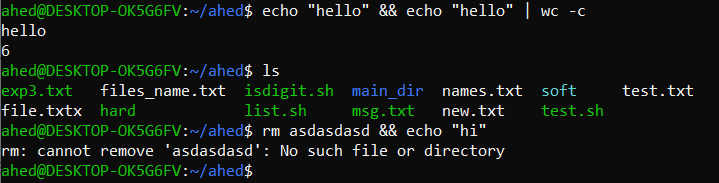
* تستخدم لما ما بدي اعمل شي اذا تحقق شرط معين عشان ما اخلي البلوك تاع if او else او حتى loop فاضي
* مثلا بدي افحص اذا arg هي 1 ما يعمل شي غير هيك يطبع thank you:



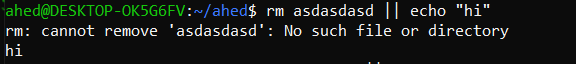


10- The && and || Constructs:

* Command1 && command2: معناها اذا كوماند1 تم بنجاح، نفذ كوماند 2 غير هيك ما تنفذ كوماند 2



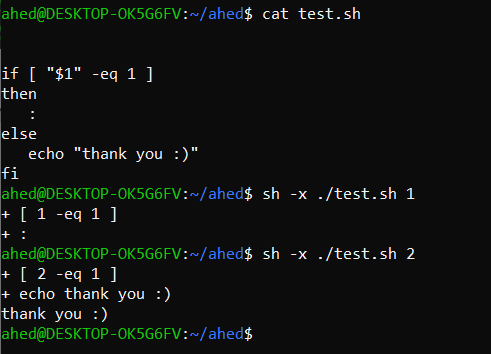
* Command1 || command2: إذا ما تم تنفيذ الكوماند 1، نفذ الكوماند 2





11- Debugging shell scripts with the-x option:

* عشان نعمل debug ل shell script نعمل الكوماند التالي :  **Sh -x ./file\_name.sh**



Tasks:)

1. Write a shell script (calc.sh) that takes 3 args: first and third one are number and the second is operator (+,-,/,\*), then do the suitable calculation based on args.

Exp: ./calc.sh 5 + 6 → output will be “sum of 5 and 6 is 11”

./calc.sh 5\*6 → output will be “mul of 5 and 6 is 30”

1. Write a shell script that checks take one argument and check if the argument is string (just letter ).
2. Write a shell script that check the memory space:

* **If the free memory space is greater than the used memory space, print:**“Free space available: [value of free space]”
* **If the free memory space equals the used memory space, print:**“No available free space”
* Note: you can free -m command to display the informations about your memory