ourdiu.com | Shared by SUPAN ROY

Operating System Lab Final Exam
Time : 35 min ; Marks : 25

1. Write a shell script program named **passwordChecker.sh**, which will identify the flaws of a password based on the condition given below.
    i. If the length of the password is less than equal 10 : it will print "*Password needs to more than 10 characters* "
    ii. If the The password doesn't contain any numbers then it will print "*Password must contain at least one number*"
    iii. If the password does not contain any special characters then it will print "*Password must contain a special character.*"

    If all the conditions mentioned are properly satisfied then it will print "All good" and append the password in a file named **StrongPassword.txt.**
    15

Input	**Output**
akiz	Password needs to more than 10 characters
akizwithten	Password must contain at least one number
akizwith10	Passwords must contain a special character.
akizwith10@	All good *[appends the password in a file named StrongPassword.txt]*

2. Create a shell script program named sumOfdigits.sh, where you will pass your last 3 digits of your id as an **argument** and it will calculate the sum of the last 3 digits of your id and print it in the terminal. **10**

Input	**Output**
213	6
920	11

ourdiu.com | Shared by SUPAN ROY

grep command extension

grep Option	Meaning	Example	Output
	Not in "[^pattern]"	`grep "[^0-9]" file.txt`	Matches lines containing non-digit characters
-i	Ignore case	`grep -i "hello" file.txt`	matches: Hello, HELLO, hello
-v	Invert match	`grep -v "error" log.txt`	all lines without "error"
-c	Count	`grep -c "fail" log.txt`	3 (if 3 lines match)
-l	File names only	`grep -l "error" *.log`	app.log
-w	Whole word	`grep -w "cat" file.txt`	matches cat but not catch
-x	Whole line	`grep -x "hello" file.txt`	hello (exact line only)
-e	Multiple patterns	`grep -e "error" -e "fail" log.txt`	lines containing error OR fail
-o	Only match	`grep -o '[0-9]' file.txt`	1 2 3 (each digit separately)

Length calculation	name="HelloWorld" len=${#name}

Ourdiu.com

condition + loop + pipeline syntax

```
echo "Enter a number:"
read num

if [ $num -eq 10 ]
then
echo "Number is 10"
elif [ $num -gt 10 ]
then
echo "Number is greater than 10"
else
echo "Number is less than 10"
fi
```

```
for ((i=1; i<=5; i++))
do
echo $i
done
```

```
while [ condition ]
do
   commands
done
```

```
a=10
b=5

echo $((a + b))
echo $((a - b))
echo $((a * b))
echo $((a / b))
```

```
cat file.txt | grep "hello"
#Shows lines containing "hello"

cat file.txt | grep "error" | wc -l
#Counts how many times "error"
appears
```

Permission	chmod [permissions] filename	
	chmod u=rwx,g=rx,o=r file.txt	chmod u+x file.sh