#### Shell Programming 2

# Aim

To write shell scripts for the following problems.

1. **To write a shell script to validate the strength of a password. Here are a few assumptions for a strong password.**

**(i) Length should be minimum of 8 characters.**

**(ii) should contain both lowercase and uppercase letters**

**(iii) should contain minimum one digit.**

**(iv) should contain minimum one special character from the set {@, \_, #, $, %, &,\*}**

**If the password doesn’t comply with any of the above conditions, then the**

**script should report it as a weak password.**

**Algorithm**

1. Start
2. Read string P from user.
3. If length of P <8 then goto step 9

End If

1. If P doesn’t contain at least 1 digit, goto step 9

End If

1. If P doesn’t contain at least 1 uppercase and 1 lowercase letter, goto step 9

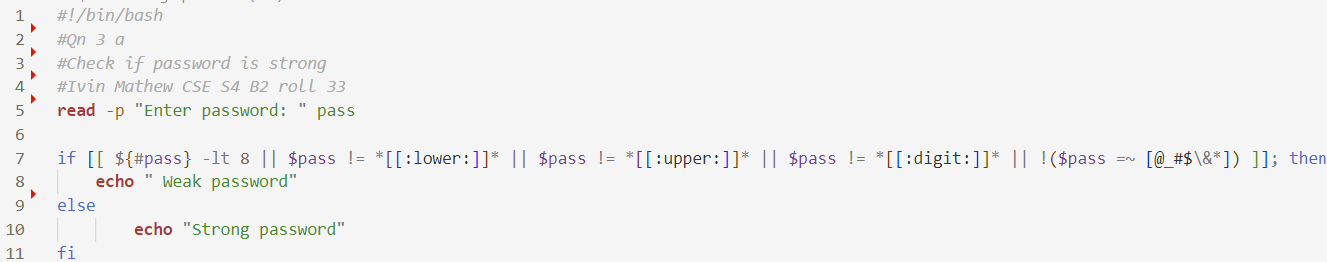
End If

1. If P doesn’t contain a special character from the following list {@, \_, #, $, %, &,\*} , then goto step 9

End If

1. Display “Strong password” and goto step 9
2. Display “Weak password”
3. Stop

**Program**

****

**Sample input and output**

****

1. **Given a CSV file, “RESULT.CSV”, containing the register number and**

**grades obtained for six subjects in an examination. First line contains**

**“register number” and subject codes. Following lines contain register**

**number of student and his/her grades. Perform a result analysis which shows**

**the following details :**

**- Number of students registered.**

**- Number of students passed in all subjects.**

**- Number of pass, fail and percentage of pass for each subject in a**

**tabular form.**

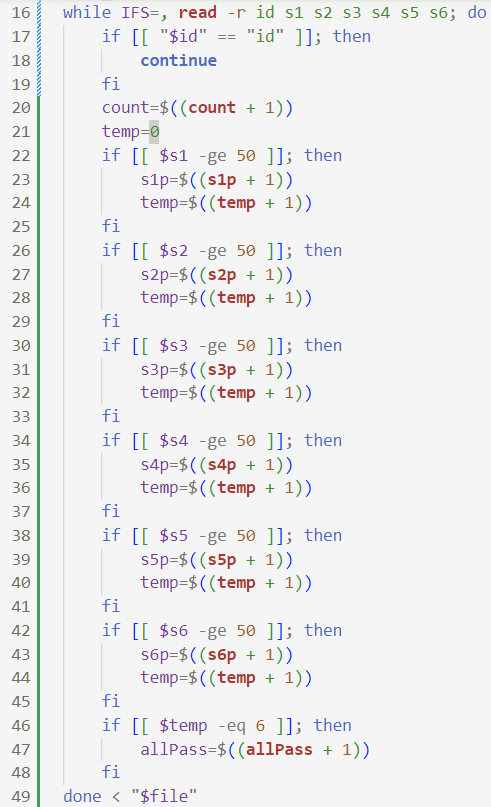
**Algorithm**

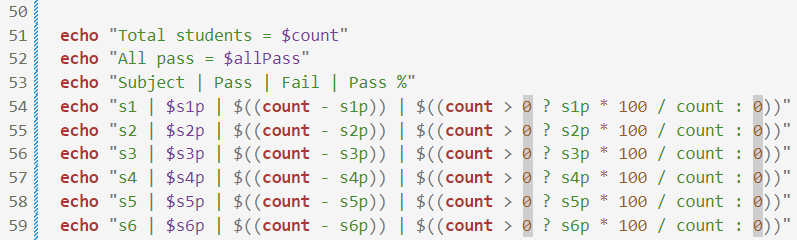
1. Start
2. Create a CSV file containing title register number and subject in the first line separated by commas. Student details must be given from the next line onwards
3. For each line in csv file
   1. If regno==”regno”
      1. Continue
   2. Increment count by 1
   3. If value of s1,s2,s3,s4,s5 and s6 are greater then 50, increment pass by 1
   4. If value of s1>50, increment s1p by 1
   5. Repeat above step for all subjects s2 to s6
4. Display count and pass, as the total no of students, and those passed in all subjects.
5. For each subject s1 to s6
   1. Display subject name
   2. Display pass = $s1p
   3. Display fail=$(count -s1p)
   4. Display pass percentage

End For

1. Stop

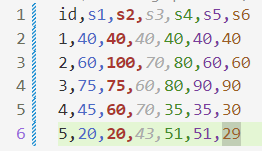
**Program** ****

****

****

**Sample input and output**

Data.csv



Output

1. **Write a script to create the command ‘summarize\_dir’ which takes a**

**directory name as command line argument and summarizes the contents in the directory. This command has to display the following:**

**(a) Total Number of Files in the directory.**

**(b) List all the extensions of files present.**

**(c) Count of files having each extension.**

**The command has to work recursively for all the subdirectories.**

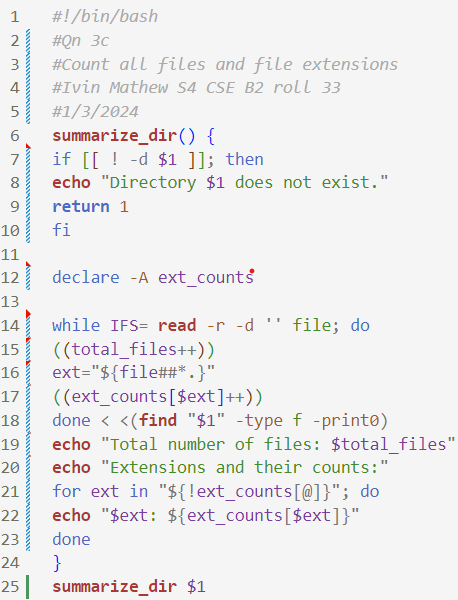
**Algorithm**

1. Start
2. Create function named summarize\_dir()
3. Read directory path from user inside function
4. If path doesn’t exist, display Invalid path. Goto step 11
5. Initialize an array to store the count of each extension
6. Iterate over all the files in the directory and its sub directories using recursion
7. For each extension

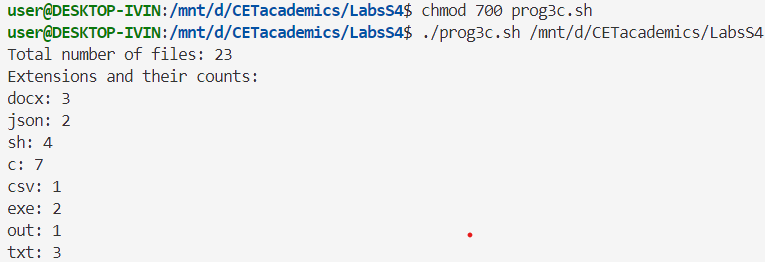
Store its name, and increment its count. Also increment count of total files

1. Display all the stored extensions and the corresponding frequency.
2. Export the function to use it as a command
3. Run the command with the file path as command line argument
4. Stop

**Program**

****

**Sample input and output**

****

**Result**

The programs have been executed and output has been verified.