

ICS 2305 SYSTEMS PROGRAMMING

TAKE-AWAY CAT

Name: AKECH DAU ATEM

RegNo: SCT211-0535/2022

Please note that as I am using macOS, some of the code may be Unix-dependent and there could be compatibility issues when running the .sh files on other systems. I have included the outputs to ensure clarity.

Thank you for your understanding.

Question 1

Write the shell script that reads any 6 numbers from the keyboard and prints their sum. If non-numbers are entered, it throws an alert to key in numbers only.

```
macbook@macbooks-MacBook-Pro shell % vi sum_numbers.sh
macbook@macbooks-MacBook-Pro shell % cat sum_numbers.sh
#!/bin/bash
# Author: Akech Dau Atem
# Student Number: SCT211-0535/2022

# Function to check if input is a valid number
is_number() {
    [[ "$1" =~ ^[0-9]+$ ]]
}

echo "Enter 6 numbers (one at a time):"
sum=0
count=0

while [ $count -lt 6 ]; do
    read -p "Enter number $((count + 1)): " input
    if is_number "$input"; then
        sum=$((sum + input))
        count=$((count + 1))
    else
        echo "Invalid input! Please enter a number."
    fi
done

echo "The sum of the 6 numbers is: $sum"

macbook@macbooks-MacBook-Pro shell % chmod +x sum_numbers.sh
macbook@macbooks-MacBook-Pro shell % ./sum_numbers.sh
Enter 6 numbers (one at a time):
Enter number 1: 12
Enter number 2: 74
Enter number 3: 45
Enter number 4: 45
Enter number 5: 24
Enter number 6: 25
The sum of the 6 numbers is: 225
macbook@macbooks-MacBook-Pro shell %
```

Question 2

Part (a): Counting Characters in a Text File

Write a shell script that reads the text in a file called Angukanayo.txt and outputs the number of characters (inclusive of white spaces) in the file.

```
[macbook@macbooks-MacBook-Pro shell % ls
sum_numbers.sh
[macbook@macbooks-MacBook-Pro shell % vi count_characters.sh
[macbook@macbooks-MacBook-Pro shell % cat count_characters.sh
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022

# Check if the file exists
file="Angukanayo.txt"

if [ -f "$file" ]; then
    # Count characters (including spaces) using wc command
    char_count=$(wc -c < "$file")
    echo "The file '$file' contains $char_count characters (including white spaces)."
else
    echo "Error: The file '$file' does not exist."
fi

[macbook@macbooks-MacBook-Pro shell % vi Angukanayo.txt
[macbook@macbooks-MacBook-Pro shell % cat Angukanayo.txt
The viral Mapangale dance, mimicking machete movements, has captivated Kenyan TikTok but sparked mixed reactions. While fans praise its creativity amid tough times, critics decry its sinister connotations, especially given rising femicide cases. Despite debate, many embrace the dance as lighthearted entertainment, highlighting Kenya's vibrant youth culture.
[macbook@macbooks-MacBook-Pro shell % ls
Angukanayo.txt      count_characters.sh  sum_numbers.sh
[macbook@macbooks-MacBook-Pro shell % chmod +x count_characters.sh

[macbook@macbooks-MacBook-Pro shell % ./count_characters.sh

The file 'Angukanayo.txt' contains      360 characters (including white spaces).
[macbook@macbooks-MacBook-Pro shell %
```

Part (b): Searching for a String in the Text File

Write a script to report if the file Angukanayo.txt contains the string Mpangale and the line number(s) on which it occurs

```
macbook@macbooks-MacBook-Pro shell % clear

macbook@macbooks-MacBook-Pro shell % cat find_string.sh
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022

# Check if the file exists
file="Angukanayo.txt"
search_string="Mpangale"

if [ -f "$file" ]; then
    # Search for the string and display line numbers
    result=$(grep -n "$search_string" "$file")

    if [ -n "$result" ]; then
        echo "The string '$search_string' is found in the file '$file' at the following line(s):"
        echo "$result"
    else
        echo "The string '$search_string' was not found in the file '$file'."
    fi
else
    echo "Error: The file '$file' does not exist."
fi

macbook@macbooks-MacBook-Pro shell % ls
Angukanayo.txt      find_string.sh
count_characters.sh  sum_numbers.sh
macbook@macbooks-MacBook-Pro shell % chmod +x find_string.sh

macbook@macbooks-MacBook-Pro shell % ./find_string.sh

The string 'Mpangale' is found in the file 'Angukanayo.txt' at the following line(s):
1:The viral Mpangale dance, mimicking machete movements, has captivated Kenyan TikTok but sparked mixed reactions. While fans praise its creativity amid tough times, critics decry its sinister connotations, especially given rising femicide cases. Despite debate, many embrace the dance as lighthearted entertainment, highlighting Kenya's vibrant youth culture.
macbook@macbooks-MacBook-Pro shell %
```

Question 3

Task

Write a shell script that, given a person's UID, tells how many times that person is logged on to the system and the time(s) of their login.

```
macbook@macbooks-MacBook-Pro shell % cat user_logins.sh
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022

# Prompt user for a UID
read -p "Enter the UID of the user: " uid

# Get the username associated with the UID using the `id` command
username=$(id -un "$uid" 2>/dev/null)

# Check if the username exists
if [ -n "$username" ]; then
    echo "Checking login details for user with UID: $uid ($username)"

    # Check how many times the user is logged in
    login_count=$(who | grep -c "^$username")

    echo "The user '$username' is currently logged in $login_count time(s)."

    # Show login times if user is logged in
    if [ "$login_count" -gt 0 ]; then
        echo "Login time(s):"
        who | grep "^$username" | awk '{print $1, $4}'
    fi
else
    echo "Error: No user found with UID $uid."
fi

macbook@macbooks-MacBook-Pro shell % id -u
501
macbook@macbooks-MacBook-Pro shell % ./user_logins.sh
Enter the UID of the user: 501
Checking login details for user with UID: 501 (macbook)
The user 'macbook' is currently logged in 2 time(s).
Login time(s):
macbook 28
macbook 29
```

Question 4

Task

Write a shell script that counts the number of empty folders in a directory (size = 0) and outputs the list of those folders along with their creation time.

```
macbook@macbooks-MacBook-Pro Desktop % vi count_empty_folders.sh
macbook@macbooks-MacBook-Pro Desktop % cat count_empty_folders.sh
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022

# Prompt user for a directory path
read -p "Enter the directory path to check: " dir_path

# Check if the directory exists
if [ -d "$dir_path" ]; then
    echo "Checking for empty folders in: $dir_path"

    # Find empty directories and store them in a temporary file
    empty_folders=$(find "$dir_path" -type d -empty)

    if [ -n "$empty_folders" ]; then
        echo "List of empty folders:"
        echo "$empty_folders"

        # Count the number of empty folders
        count=$(echo "$empty_folders" | wc -l)
        echo "Total number of empty folders: $count"

        # Display creation times
        echo "Creation times of empty folders:"
        for folder in $empty_folders; do
            # Use stat to retrieve creation time
            creation_time=$(stat -f "%SB" -t "%Y-%m-%d %H:%M:%S" "$folder" 2>/dev/null)
            echo "$folder - Created on: $creation_time"
        done
    else
        echo "No empty folders found in the directory."
    fi
else
    echo "Error: The directory path '$dir_path' does not exist."
fi
```

1. Directory Check:

The script checks if the provided directory exists using `-d`.

2. Find Empty Folders:

The `find` command searches for directories (`-type d`) that are empty (`-empty`).

3. Count and List Empty Folders:

If empty folders are found, they are listed, and their total count is displayed using `wc -l`.

4. Retrieve Creation Times:

The `stat` command is used to retrieve the creation time of each empty folder (macOS-specific).

```

macbook@macbooks-MacBook-Pro Desktop % chmod +x count_empty_folders.sh

macbook@macbooks-MacBook-Pro Desktop % ./count_empty_folders.sh

Enter the directory path to check: /Users/macbook/Desktop
Checking for empty folders in: /Users/macbook/Desktop
List of empty folders:
/Users/macbook/Desktop/pop
/Users/macbook/Desktop/SWMS/env/include/python3.12
/Users/macbook/Desktop/processes/.dist
/Users/macbook/Desktop/file_management/.dist
/Users/macbook/Desktop/env/include/python3.12
/Users/macbook/Desktop/New Folder With Items/portfolio/.git/objects/info
/Users/macbook/Desktop/New Folder With Items/portfolio/.git/refs/tags
/Users/macbook/Desktop/unix
/Users/macbook/Desktop/.venv/include/python3.12
/Users/macbook/Desktop/stack/.dist
/Users/macbook/Desktop/c/.dist
Total number of empty folders:      11
Creation times of empty folders:
/Users/macbook/Desktop/pop - Created on: 2024-11-26 15:22:40
/Users/macbook/Desktop/SWMS/env/include/python3.12 - Created on: 2024-11-18 10:01:59
/Users/macbook/Desktop/processes/.dist - Created on: 2024-11-21 12:54:38
/Users/macbook/Desktop/file_management/.dist - Created on: 2024-10-28 09:08:46
/Users/macbook/Desktop/env/include/python3.12 - Created on: 2024-11-18 09:59:27
/Users/macbook/Desktop/New - Created on:
Folder - Created on:
With - Created on:
Items/portfolio/.git/objects/info - Created on:
/Users/macbook/Desktop/New - Created on:
Folder - Created on:
With - Created on:
Items/portfolio/.git/refs/tags - Created on:
/Users/macbook/Desktop/unix - Created on: 2024-11-26 05:21:57
/Users/macbook/Desktop/.venv/include/python3.12 - Created on: 2024-06-19 14:37:31
/Users/macbook/Desktop/stack/.dist - Created on: 2024-10-15 06:41:16
/Users/macbook/Desktop/c/.dist - Created on: 2024-10-11 18:15:58
macbook@macbooks-MacBook-Pro Desktop % █

```

Question 5

Task

Write a shell script that will convert all .jpg files in a given directory into .tiff files. The user should be prompted to enter the file path. The output after conversion should give the number of images converted.

```
shell — -zsh — 158x38

if [ -d "$dir_path" ]; then
    echo "Checking for .jpg files in: $dir_path"

    # Find all .jpg files and convert them to .tiff
    jpg_files=$(find "$dir_path" -type f -iname "*.jpg")

    # Initialize counter for converted files
    converted_count=0

    # Loop through each .jpg file found
    while IFS= read -r file; do
        # Get the filename without the extension
        base_name=$(basename "$file" .jpg)

        # Convert the .jpg file to .tiff
        convert "$file" "$dir_path/$base_name.tiff"

        if [ $? -eq 0 ]; then
            echo "Converted: $file to $dir_path/$base_name.tiff"
            ((converted_count++))
        else
            echo "Error converting: $file"
        fi
    done <<< "$jpg_files"

    # Output the number of images converted
    echo "Total number of images converted: $converted_count"
else
    echo "Error: The directory '$dir_path' does not exist."
fi

macbook@macbooks-MacBook-Pro shell % ./convert_images.sh
Enter the directory path containing .jpg files: /Users/macbook/Desktop/unix
Checking for .jpg files in: /Users/macbook/Desktop/unix
Converted: /Users/macbook/Desktop/unix/jpeg-home.jpg to /Users/macbook/Desktop/unix/jpeg-home.tiff
Converted: /Users/macbook/Desktop/unix/aic-home.jpg to /Users/macbook/Desktop/unix/aic-home.tiff
Total number of images converted: 2
macbook@macbooks-MacBook-Pro shell %
```

Question 6

Task:

Write a bash script that scans the network for hosts attached to an IP address. The script should show if the host is up, indicating "Yes" if up.


```
macbook@macbooks-MacBook-Pro shell % cat network_host_scanner.sh
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022

# Prompt the user to enter the IP address or network range
read -p "Enter the IP address or network range to scan (e.g., 192.168.1.0/24): "
ip_range

# Check if the input is not empty
if [ -z "$ip_range" ]; then
    echo "Error: No IP address or network range entered."
    exit 1
fi

# Notify the user that scanning has started
echo "Scanning the network for active hosts on $ip_range..."

# Use nmap to scan for active hosts
nmap -sn $ip_range | while read -r line; do
    # Check for lines indicating a host is up
    if [[ $line == *"Nmap scan report for"* ]]; then
        host=$(echo $line | awk '{print $5}')
        echo "Host: $host - Status: Yes (Up)"
    fi
done

macbook@macbooks-MacBook-Pro shell % ipconfig getifaddr en1
192.168.0.105
macbook@macbooks-MacBook-Pro shell % ./network_host_scanner.sh
Enter the IP address or network range to scan (e.g., 192.168.1.0/24): 192.168.0.
105
Scanning the network for active hosts on 192.168.0.105...
Host: 192.168.0.105 - Status: Yes (Up)
macbook@macbooks-MacBook-Pro shell %
```

Question 7

Task:

The Department Chairman of Computing has requested a mail merge service to invite students to a webinar. Write a shell script that reads a text file with names and emails, then sends personalized invitations.

```
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022
```

```

# Prompt the user for the file containing the names and emails
read -p "Enter the file name containing the names and emails (e.g., students_list.txt): " file_name

# Check if the file exists
if [ ! -f "$file_name" ]; then
echo "Error: File '$file_name' does not exist."
exit 1
fi

# Read the file and send invitations
while IFS=',' read -r name email; do
echo "Sending invitation to $name <$email>..."
mail -s "Invitation to JhubAfrica Webinar" "$email" <<EOF
Dear $name,

You are invited to our 21st Open Webinar on Skills Optimization by JhubAfrica.
**Date:** Friday, 29th November
**Time:** 5:00 P.M.
**Zoom Link:** https://youtu.be/ONVTA7LKMIIs

We will also broadcast the webinar on our YouTube channel.
**YouTube Link:** https://youtu.be/ONVTA7LKMIIs

Looking forward to your participation.

Best regards,
Department of Computing
EOF
done < "$file_name"

echo "All invitations sent!"

```

Question 8

Task

Write a bash script that checks the available disk space on your drive (e.g., C:) and alerts the user with a message indicating the amount of space available in megabytes (MB).

```
Terminal Shell Edit View Window Help
[macbook@macbooks-MacBook-Pro shell % pwd
/Users/macbook/Desktop/shell
[macbook@macbooks-MacBook-Pro shell % vi check_disk_space.sh
[macbook@macbooks-MacBook-Pro shell % cat check_disk_space.sh
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022

# Check available disk space on the root directory
disk_space=$(df -m / | awk 'NR==2 {print $4}')

# Display an alert with the available space in MB
echo "The space available on the drive is ${disk_space}MB."

macbook@macbooks-MacBook-Pro shell % chmod +x check_disk_space.sh

macbook@macbooks-MacBook-Pro shell % ./check_disk_space.sh

The space available on the drive is 160901MB.
macbook@macbooks-MacBook-Pro shell %
```

Question 9

Task

Part (a): Using the wget command, download two files:

1. An example TXT file:
<https://github.com/yesinteractive/dadjokes/blob/master/controllers/jokes.txt>
2. An example JSON file: <https://gist.github.com/rominirani/8235702#file-employees-json>

```
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022

# URLs to download
```

```
txt_url="https://raw.githubusercontent.com/yesinteractive/dadjokes/master/controllers/jokes.txt"
json_url="https://gist.githubusercontent.com/rominirani/8235702/raw/6820f70bd9187f9db51d97e089759fe92d8c1f1d/employees.json"

# Download the TXT file
wget -O jokes.txt "$txt_url"

# Download the JSON file
wget -O employees.json "$json_url"

# Verify if the files were downloaded
if [ -f "jokes.txt" ] && [ -f "employees.json" ]; then
echo "Files downloaded successfully."
else
echo "Error: Failed to download one or both files."
fi
```

Download Files:

The wget -O command downloads the file and saves it with the specified name (jokes.txt and employees.json).

Verification:

The script checks if both files exist using [-f].

Part (b): Show Lines 2 to 7 of jokes.txt

Use the sed command to display lines 2 through 7:

```
shell — -zsh — 61x38
[macbook@macbooks-MacBook-Pro shell % pwd
/Users/macbook/Desktop/shell
[macbook@macbooks-MacBook-Pro shell % vi show_lines.sh
macbook@macbooks-MacBook-Pro shell % chmod +x show_lines.sh

macbook@macbooks-MacBook-Pro shell % ./show_lines.sh

Displaying lines 2 to 7 from jokes.txt:
I burned 2000 calories today<>I left my food in the oven for
too long.
I startled my next-door neighbor with my new electric power t
ool. <>I had to calm him down by saying "Don't worry, this is
just a drill!"
I broke my arm in two places. <>My doctor told me to stop goi
ng to those places.
I quit my job at the coffee shop the other day. <>It was just
the same old grind over and over.
I never buy anything that has Velcro with it...<>it's a total
rip-off.
I used to work at a soft drink can crushing company...<>it wa
s soda pressing.
macbook@macbooks-MacBook-Pro shell %
```

Part (c): Create an MD5 Checksum for Both Files

This script generates an **MD5 checksum** for jokes.txt and employees.json, saving the checksums to a file named MD5SUM

```
macbook@macbooks-MacBook-Pro shell % pwd
/Users/macbook/Desktop/shell
macbook@macbooks-MacBook-Pro shell % vi generate_md5.sh
macbook@macbooks-MacBook-Pro shell % cat generate_md5.sh
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022

# Check if both files exist
if [ -f "jokes.txt" ] && [ -f "employees.json" ]; then
    echo "Generating MD5 checksum for jokes.txt and employees.json..."

    # Generate checksums and save to MD5SUM file
    md5 jokes.txt > MD5SUM
    md5 employees.json >> MD5SUM

    echo "MD5 checksums saved to MD5SUM file."
else
    echo "Error: One or both files not found. Please ensure jokes.txt and
employees.json are downloaded."
fi

macbook@macbooks-MacBook-Pro shell % chmod +x generate_md5.sh

macbook@macbooks-MacBook-Pro shell % ./generate_md5.sh

Generating MD5 checksum for jokes.txt and employees.json...
MD5 checksums saved to MD5SUM file.
macbook@macbooks-MacBook-Pro shell %
```

Part (d): Archive Both Files as Johnnie.tar.gz

Here's a script to create a tarball archive (Johnnie.tar.gz) containing both jokes.txt and employees.json.

```

[macbook@macbooks-MacBook-Pro shell % vi archive_files.sh
[macbook@macbooks-MacBook-Pro shell % cat archive_files.sh
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022

# Define the files to archive
jokes_file="/Users/macbook/Desktop/shell/jokes.txt"
json_file="/Users/macbook/Desktop/shell/employees.json"
archive_name="Johnnie.tar.gz"

# Check if both files exist
if [ -f "$jokes_file" ] && [ -f "$json_file" ]; then
    echo "Archiving jokes.txt and employees.json into $archive_name..."

    # Create the tarball archive
    tar -czf "$archive_name" -C "/Users/macbook/Desktop/shell" "jokes.txt" "employees.json"

    # Confirm the archive was created
    if [ -f "$archive_name" ]; then
        echo "Files successfully archived as $archive_name."
    else
        echo "Error: Failed to create the archive."
    fi
fi
else
    echo "Error: One or both files not found. Please ensure jokes.txt and employees.json exist."
fi

macbook@macbooks-MacBook-Pro shell % chmod +x archive_files.sh

macbook@macbooks-MacBook-Pro shell % ./archive_files.sh

Archiving jokes.txt and employees.json into Johnnie.tar.gz...
Files successfully archived as Johnnie.tar.gz.
macbook@macbooks-MacBook-Pro shell % █

```

Part (e): Delete Both Files

This script will delete the downloaded files (jokes.txt and employees.json) after they have been archived.

```
macbook@macbooks-MacBook-Pro shell % vi delete_files.sh
macbook@macbooks-MacBook-Pro shell % cat delete_files.sh
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022

# Define the files to delete
jokes_file="/Users/macbook/Desktop/shell/jokes.txt"
json_file="/Users/macbook/Desktop/shell/employees.json"

# Check if both files exist
if [ -f "$jokes_file" ] && [ -f "$json_file" ]; then
    echo "Deleting jokes.txt and employees.json..."

    # Delete the files
    rm "$jokes_file" "$json_file"

    # Verify if the files were deleted
    if [ ! -f "$jokes_file" ] && [ ! -f "$json_file" ]; then
        echo "Files successfully deleted."
    else
        echo "Error: Failed to delete the files."
    fi
else
    echo "Error: One or both files not found. Please ensure jokes.txt and
employees.json exist."
fi

macbook@macbooks-MacBook-Pro shell % chmod +x delete_files.sh

macbook@macbooks-MacBook-Pro shell % ./delete_files.sh

Deleting jokes.txt and employees.json...
Files successfully deleted.
macbook@macbooks-MacBook-Pro shell %
```

Part (f): Recover or Restore the Files from the Archive

This script will restore the files (jokes.txt and employees.json) from the Johnnie.tar.gz archive that was created in **Part (d)**.


```
macbook@macbooks-MacBook-Pro shell % vi restore_files.sh
macbook@macbooks-MacBook-Pro shell % cat restore_files.sh
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022

# Define the archive file and restore destination
archive_name="Johnnie.tar.gz"
restore_dir="/Users/macbook/Desktop/shell"

# Check if the archive exists
if [ -f "$archive_name" ]; then
    echo "Restoring files from $archive_name..."

    # Extract the files from the archive
    tar -xzf "$archive_name" -C "$restore_dir"

    # Verify if the files were restored
    if [ -f "$restore_dir/jokes.txt" ] && [ -f "$restore_dir/employees.json" ]; then
        echo "Files successfully restored."
    else
        echo "Error: Failed to restore one or both files."
    fi
else
    echo "Error: $archive_name not found. Please ensure the archive exists."
fi

macbook@macbooks-MacBook-Pro shell % chmod +x restore_files.sh

macbook@macbooks-MacBook-Pro shell % ./restore_files.sh

Restoring files from Johnnie.tar.gz...
Files successfully restored.
macbook@macbooks-MacBook-Pro shell %
```

Part (g): Compare MD5 Checksums

This script compares the MD5 checksums of the restored files (jokes.txt and empl

```
shell — -zsh — 86x38
#!/bin/bash
# Name: Akech Dau Atem
# RegNo: SCT211-0535/2022

# Define the files to compare
jokes_file="/Users/macbook/Desktop/shell/jokes.txt"
json_file="/Users/macbook/Desktop/shell/employees.json"
checksum_file="MD5SUM"

# Check if the MD5SUM file and the restored files exist
if [ -f "$checksum_file" ] && [ -f "$jokes_file" ] && [ -f "$json_file" ]; then
    echo "Comparing MD5 checksums..."

    # Generate the MD5 checksums for the restored files
    md5 -r "$jokes_file" > MD5CHECK
    md5 -r "$json_file" >> MD5CHECK

    # Compare the original MD5SUM file with the new checksums
    if diff "$checksum_file" MD5CHECK > /dev/null; then
        echo "Checksums match: The files are intact."
    else
        echo "Checksums do not match: The files may have been altered."
    fi

    # Clean up the temporary MD5CHECK file
    rm MD5CHECK
else
    echo "Error: MD5SUM file or one or both of the files do not exist."
fi

macbook@macbooks-MacBook-Pro shell % chmod +x compare_md5.sh

macbook@macbooks-MacBook-Pro shell % ./compare_md5.sh

Comparing MD5 checksums...
Checksums do not match: The files may have been altered.
macbook@macbooks-MacBook-Pro shell %
```