This is the script ( Kenscript.sh) up to the point where option is malfunctioning

#!/bin/bash

clear # clear the screen

echo -e "Name of Script: Kenscript.sh"

echo -e "By: KA"

echo -e "Date: 4/14/2023"

echo -e "Purpose: Shell Script Test"

echo -e "Command Line: Kenscript.sh"

echo

echo

**while** true; **do**

# Print the menu

echo "Please select an option:"

echo "1. Show current date and time"

echo "2. Show first letter of your name"

echo "3. Show line from 12 Days of Christmas song"

echo "4. Show sum of two digits that make up your age"

echo "5. Do anything you want!"

echo "0. Exit program"

read -p "Enter your choice: " menu\_choice

# Process the user's choice

**case** $menu\_choice **in**

1)

# Show the current date and time

date

;;

2)

# Show the first letter of the user's name

read -p "Enter your first name: " first\_name

echo "The first letter of your name is ${first\_name:0:1}"

;;

3)

# Show a line from the 12 Days of Christmas song

read -p "Enter a number between 1 and 12: " day\_number

lyrics=(

"a partridge in a pear tree"

"two turtle doves"

"three French hens"

"four calling birds"

"five golden rings"

"six geese a-laying"

"seven swans a-swimming"

"eight maids a-milking"

"nine ladies dancing"

"ten lords a-leaping"

"eleven pipers piping"

"twelve drummers drumming"

)

**if** ((day\_number >= 1 && day\_number <= 12)); **then**

echo "On the $day\_number day of Christmas my true love gave to me: ${lyrics[day\_number-1]}"

**else**

echo "Invalid day number."

**fi**

;;

4)

# Main logic

read -p "Enter your age: " value

**if** ! [[ $value =~ ^[0-9]+$ ]]; **then**

echo "Error: Invalid input. Please enter a positive integer."

exit 1

**fi**

half=$((value/2))

random\_num=$((RANDOM % half + 1))

**if** [[ $((value - random\_num)) =~ ^[0-9]+$ ]]; **then**

echo "The two numbers that add up to your age are $random\_num and $((value-random\_num))."

exit 0

**fi**

;;

5)

# Perform any command(s) the user wants

read -p "Enter any command(s) you want to run: " user\_commands

eval "$user\_commands"

;;

0)

# Exit the program

exit 0

;;

\*)

# Invalid choice

echo "Invalid choice. Please try again."

;;

**esac**

# Wait for user input before clearing the screen

read -n 1 -s -r -p "Press any key to continue..."

clear

**done**