# Go to the file's current directory

cd "$(dirname "${BASH\_SOURCE[0]}")"

# Path to odd solution file. Change to your need.

odd\_path="/OddZone/odd\_answer.sh"

# Path to even solution file. Change to your need.

even\_path="/EvenZone/even\_answer.sh"

# Stores the requested user action number

action=0

# Displays the given text as colored, then returns to white text.

# First parameter is the text to be displayed.

# Second parameter is the special color character.

display\_colored() {

# Data

local txt=$1

# Color

local color=$2

# Display info

echo -en "$color$txt$(tput setaf 7)"

}

# Displays an error message (red).

# Takes text as input.

display\_err() {

# Display colored information

local err=$1

# Display colored information (red)

display\_colored "\n$err\n" "$(tput setaf 1)"

}

# Displays a success message, colored green.

# Takes text as input.

display\_success() {

# Input message

local msg=$1

# True removes new line characters

local rmv\_pad=${2:-false}

if [ $rmv\_pad != true ]; then

msg="\n$msg\n"

fi

# Display colored information (green).

display\_colored "$msg" "$(tput setaf 2)"

}

# Function to display the menu & set the chosen user action

display\_menu() {

# Empty new line

echo ""

echo "1- Odd"

echo "2- Even"

echo "3- Exit"

# Empty new line

echo ""

# Read user choice from console

read action

# Check if the user input is a valid integer using regex;

# if check does not pass, set the choice to -1.

if ! { [[ "$action" =~ ^[0-9]+$ ]]; }; then

action=-1

fi

}

# Keep on taking user input till exiting

while true; do

# Call the display menu, fill take action input

display\_menu

# Switch user action

case $action in

1) # Run odd

./$odd\_path

;;

2) # Run even

./$even\_path

;;

3) # Exit

display\_success "Exiting the script. Goodbye!"

exit 0

;;

\*)

display\_err "Invalid choice. Please enter a number between 1 and 3."

;;

esac

done