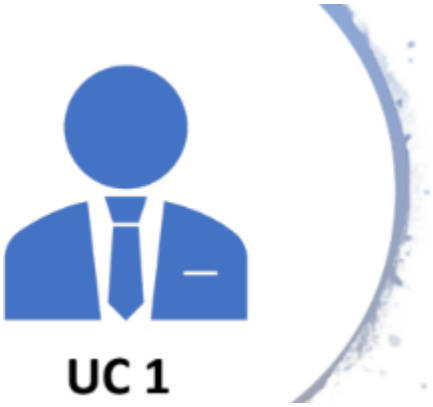
Main merged files:

```
  GNU nano 6.4                                  flipCoinSimulator.sh
echo "Welcome to Flip Coin Simulator !!!"

flip=$((RANDOM % 2))
heads=0
tails=0
goal=21

while [ $heads -lt $goal ] && [ $tails -lt $goal ]; do
  flip=$((RANDOM % 2))

  if [ $flip -eq 0 ]; then
    heads=$((heads + 1))
  else
    tails=$((tails + 1))
  fi

  if [ $heads -eq $tails ] && [ $heads -ge $goal ]; then
    while [ $((heads - tails)) -lt 2 ] && [ $((tails - heads)) -lt 2 ]; do
      flip=$((RANDOM % 2))

      if [ $flip -eq 0 ]; then
        heads=$((heads + 1))
      else
        tails=$((tails + 1))
      fi
    done
  fi
done

echo "Heads: $heads"
echo "Tails: $tails"

if [ $heads -eq $tails ]; then
  echo "It's a tie!"
elif [ $heads -gt $tails ]; then
  echo "Heads wins by $((heads - tails)) points"
else
  echo "Tails wins by $((tails - heads)) points"
fi
```

```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo4 (main)
$ sh flipCoinSimulator.sh
Welcome to Flip Coin Simulator !!!
Heads: 7
Tails: 21
Tails wins by 14 points
```



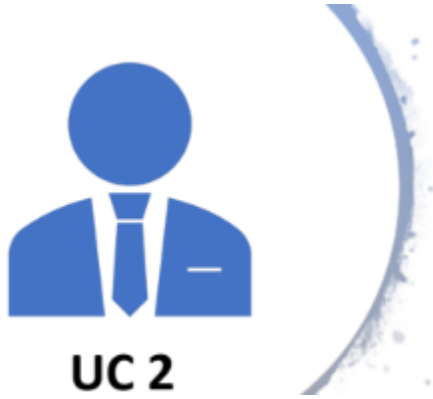
UC 1

As a Simulator start with Flipping a Coin to Display Heads or Tails as winner

- Use ((RANDOM)) to find Heads or Tails
- Name the file flipCoinSimulator.sh

```
  GNU nano 6.4                                  flipCoinSimulator.sh
flip=$((RANDOM % 2))

if [ $flip -eq 0 ]; then
  echo "Heads"
else
  echo "Tails"
fi
```

```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo4 (uc1)
$ sh flipCoinSimulator.sh
Tails
```

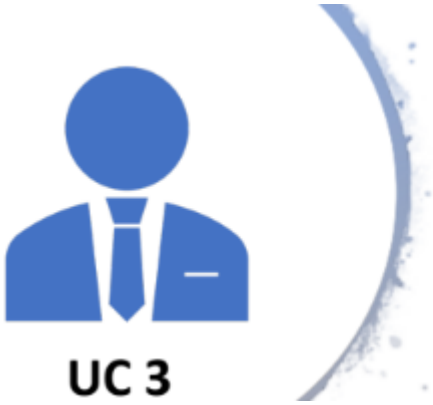


UC 2

As a simulator, loop through Multiple times of flipping a coin and show number of times head and tail has won

```
  GNU nano 6.4                                  flipCoinSimulator.sh
heads=0
tails=0
flips=10

for ((i=1; i<=flips; i++)); do
  flip=$((RANDOM % 2))

  if [ $flip -eq 0 ]; then
    heads=$((heads + 1))
  else
    tails=$((tails + 1))
  fi
done

echo "Heads: $heads"
echo "Tails: $tails"
```

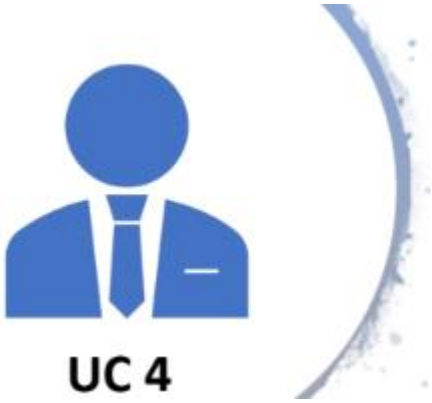
```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo4 (uc2)
$ sh flipCoinSimulator.sh
Heads: 5
Tails: 5
```



Modify the earlier UC 2 to continue till either of them have won 21 times. Show if it's a Win or Tie. If Win then who won and by how much

```
  GNU nano 6.4                                  flipCoinSimulator.sh
heads=0
tails=0
goal=21

while [ $heads -lt $goal ] && [ $tails -lt $goal ]; do
  flip=$((RANDOM % 2))

  if [ $flip -eq 0 ]; then
    heads=$((heads + 1))
  else
    tails=$((tails + 1))
  fi
done

echo "Heads: $heads"
echo "Tails: $tails"

if [ $heads -eq $tails ]; then
  echo "It's a tie!"
elif [ $heads -gt $tails ]; then
  echo "Heads wins by $((heads - tails)) points"
else
  echo "Tails wins by $((tails - heads)) points"
fi
```

```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo4 (uc3)
$ sh flipCoinSimulator.sh
Heads: 13
Tails: 21
Tails wins by 8 points
```



UC 4

Extend UC 3 to ensure if its tie then the game continues till the difference of minimum 2 points is achieved

```
  GNU nano 6.4                                  flipCoinSimulator.sh
heads=0
tails=0
goal=21

while [ $heads -lt $goal ] && [ $tails -lt $goal ]; do
  flip=$((RANDOM % 2))

  if [ $flip -eq 0 ]; then
    heads=$((heads + 1))
  else
    tails=$((tails + 1))
  fi

  if [ $heads -eq $tails ] && [ $heads -ge $goal ]; then
    while [ $((heads - tails)) -lt 2 ] && [ $((tails - heads)) -lt 2 ]; do
      flip=$((RANDOM % 2))

      if [ $flip -eq 0 ]; then
        heads=$((heads + 1))
      else
        tails=$((tails + 1))
      fi
    done
  fi
done

echo "Heads: $heads"
echo "Tails: $tails"

if [ $heads -eq $tails ]; then
  echo "It's a tie!"
elif [ $heads -gt $tails ]; then
  echo "Heads wins by $((heads - tails)) points"
else
  echo "Tails wins by $((tails - heads)) points"
fi
```

```
Shri@PRODUCTIVITY-4 MINGW64 ~/Testing_Bridge/repoPortal/repo4 (uc4)
$ sh flipCoinSimulator.sh
Heads: 21
Tails: 14
Heads wins by 7 points
```