오픈소스SW개론 2024-OSS-Project1 과제

컴퓨터공학과 12211823 이현서

#!/bin/bash

if [ "$#" -ne 3 ]; then

echo "usage: $0 file1 file2 file3"

exit 1

else

echo "usage: $0 $1 $2 $3"

fi

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*OSS1 - Project1\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

echo "\* StudentID : 12211823 \*"

echo "\* Name : Hyeonseo Lee \*"

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

ans=0

until [ $ans -eq 7 ]

do

echo "[MENU]"

echo "1. Get the data of Heung-Min Son’s Current Club, Appearance, Goals, Assists in players.csv"

echo "2. Get the team data to enter a league position in teams.csv"

echo "3. Get the Top-3 Attendance matches in mateches.csv"

echo "4. Get the team's league position and team's top scorer in teams.csv & players.csv"

echo "5. Get the modified format of date\_GMT in matches.csv"

echo "6. Get the data of the winning team by the largest difference on home stadium in teams.csv & matches.csv"

echo "7. Exit"

read -p "Enter your CHOICE (1~7) : " choice

case $choice in

1)

read -p "Do you want to get the Heung-Min Son's data? (y/n) : " key

if [ "$key" = "y" ]; then

cat players.csv | awk -F',' '$1=="Heung-Min Son" {print "Team: ", $4, "Appearance: ", $6, "Goal: ", $7, "Assist: ", $8}'

fi

;;

2)

read -p "What do you want to get the team data of league\_position[1~20] : " key

cat teams.csv | awk -F',' -v key="$key" '$2==key {print $6, " ", $1, " ", $2/($2+$3+$4)}'

;;

3)

read -p "Do you want to know Top-3 attendance data? (y/n): " key

if [ "$key" = "y" ]; then

cat matches.csv | awk -F',' '{print $3, " vs ", $4, " (", $1, ")\n", $2, " ", $7}' | sort -r -k2n | head -n 3

fi

;;

4)

read -p "Do you want to get each team's ranking and the highest-scoring player? (y/n) : " key

if [ "$key" = "y" ]; then

IFS=,

teams=()

while read -r score name; do

teams+=("$2, $1")

done < teams.csv

IFS=$'\n'

sorted\_teams=($(sort -r -t, -k2n <<<"${teams[\*]}"))

unset IFS

rank=1

for team in "${sorted\_teams[@]}"; do

IFS=,

read -r score name <<< "$team"

top\_scorer=$(awk -F',' -v team\_name="$name" '$4==team\_name {print $1, $7}' players.csv | sort -r -k2n | head -n 1)

printf "%d. %20s %s\n" "$rank" "$name" "$top\_scorer"

((rank++))

unset IFS

done

fi

;;

5)

read -p "Do you want to modify the format of date? (y/n): " key

if [ "$key" = "y" ]; then

cat $1 matches.csv | head -n 10 | sed -E 's/([0-9]{4})-([0-9]{2})-([0-9]{2}) ([0-9]{2}):([0-9]{2}):([0-9]{2})/\(am\|pm\)/\[\1\/\2\/\3 \4:\5\6]/g; s/ ([0-9]{2}):([0-9]{2})([ap]m)/ \1:\2\3/g'

fi

;;

6)

list=($(cat teams.csv | awk -F',' '{print $1}'))

select name in "${list[@]}"; do

if [[ -n $name ]]; then

echo -e "Enter your team number: ${name}\n\n"

cat matches.csv | awk -F',' -v name=“$name” '$3==name{print $1, "\n", $3, " ", $5, " vs ", $6, " ", $4}' matches.csv

break

fi

done

;;

7) echo "Bye"

exit 0

;;

esac

echo

done

7번 선택지가 나오기 전까지 MENU가 계속 반복되어야 하므로 until 구문을 활용하여 변수 값에 미리 0을 넣어준 후 7일 때 탈출 하도록 반복문 조건을 걸어주었다.

read 명령어에 -p를 하여 변수 값을 바로 받을 수 있게 하였다.

case문을 활용하여 변수 값이 1~7인 경우 각각 처리할 수 있게 하였다.

1인 경우 read -p를 통해 변수에 값을 받아오고 그 값을 if문을 통해 y일 때 해당 bash 명령어를 수행하도록 하였다.

players.csv파일 내용을 다 받아오고 그걸 파이프라인을 통해 awk 로 전달하여 ,를 구분자로써 제거해주고 Heung-Min Son 문자열을 첫번째 attribute의 값들과 비교한 후 일치하는 tuple이 있으면 해당 라인의 4번 컬럼과 6번컬럼, 7, 8번 컬럼들을 출력하도록 하였다.

2인 경우 key 변수 값에 read -p로 바로 값을 받아오고 cat으로 teams.csv파일을 받아온 후 구분자, 를 없애주고 key 변수값을 해당 명령어 내에서 선언해주고 그 변수 값이 2번째 컬럼에 있는 경우의 6번째 컬럼과 1번째 컬럼, 그리고 2번째컬럼/(2번째 컬럼 + 3번째 컬럼 + 4번째 컬럼) 값을 출력해준다

3인 경우 key값을 변수로 바로 받아오고 key값이 y인 경우 if문이 실행된다. cat에서 matches.csv파일을 받아오고 awk에서 -F가 구분자 ,를 제거해주고 3번째 컬럼, vs문자열, 4번째 컬럼, (문자열, 1번째 컬럼, )문자열과 엔터키, 2번째 컬럼, 7번째 컬럼을 파이프라인을 통해 sort로 보내고 이 값들 중 2번째 컬럼 값을 내림차순으로 정렬하여 상위 3개만 출력한다.

4인 경우 key에 변수 값을 받아오고 문자열 y인 경우 구분자 IFS에 ,을 넣어주고 출력순서 값을 나타내는 num변수를 0으로 초기화 시켜준다.

teams.csv파일에서 팀 이름과 점수를 읽어오고 teams배열에 저장한다. while문에서 팀 이름과 점수를 내림차순으로 정렬해서 teams배열에 저장한다. 그리고 rank값을 1로 초기화 한 후 for문에서 각 팀의 순위와 최고점수 선수를 출력한다.

그리고 IFS 변수를 해제시켜준다.

5인 경우 key 값에 변수를 바로 받아오고 matches.csv 파일에서 상위 10개 값을 받아와서 파이프라인으로 sed 로 넘겨주고 [0-9]범위 내에서 4개 값, - 출력 이렇게 해서 각 값들을 변환시켜준다.

6인경우 select문으로 teams.csv파일의 1번째 컬럼 값으로 메뉴를 구성해준다. if문으로 파일 내에 있는 값이 들어왔을 때만 처리하고 한번만 처리하도록 구조를 짜주고 cat과 파이프라인, awk를 사용하여 각 컬럼 값들을 출력한다

7인 경우 Bye를 출력하며 exit 0을 통해 종료상태값을 0으로 해준다.

case문 밖에 echo를 넣어주어 각 반복문 마다 엔터키를 넣어주었다.