OSS Project1

12201713 김인권

파일을 실행시킬 때 입력으로 들어오는 인자인 .csv파일 3개가 들어오지 않으면 에러메세지를 출력하고 실행종료시키기 위해 if [ $# -ne 3 ]으로 인자의 갯수가 3개가 아닐경우 에러메세지를 출력

예시 출력에 맞게 echo를 이용하여 studentID, Name을 출력

큰 틀:

사용자에게 7가지의 메뉴를 출력해주고 알맞는 숫자가 입력되면 해당번째 메뉴에 대한 내용을 출력 => until do done문 이용

stop변수를 N으로 초기화한 다음, 사용자가 7을 입력하면 stop을 Y로 바꾸어 until문을 탈출하게 하여 실행종료시킴.

메뉴의 갯수가 7가지 이므로 case문 이용, 사용자가 입력한 숫자에 해당하는 메뉴를 수행하고 7을 입력하기 전까지 계속 사용자에게 입력을 요구하기 때문에 until문 안에 case문을 사용

사용자에게 출력해주는 메뉴는 echo를 이용하여 출력

사용자가 1~7사이의 숫자가 아닌 다른 입력이 들어오는 경우가 발생가능하기 때문에 예외처리가 필요 case문에 \*)의 case를 추가하여 1~7이 아닌 수가 입력되었을 때 에러메세지를 출력하고 다시 입력 받는다.

=>코드로 변환

stop="N"

until [ "$stop" = "Y" ]

do

echo ""

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 matches.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) :" menuNum

case "$menuNum" in

1)

;;

2)

;;

3)

;;

4)

;;

5)

;;

6)

;;

7)

echo "Bye!"

stop="Y"

;;

\*)

echo "Wrong Input..Please insert 1~7"

;;

esac

done

-case 1

손흥민의 데이터를 입력받을지 물어보고 y 또는 n을 입력받기

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

check변수가 y인 경우에만 손흥민의 데이터 출력

=> if [ "$check" = "y" ]

then

fi

players.csv에서 콤마를 구분자로 첫번째 필드가 Heung-Min son이면 출력양식에 맞게 해당 필드를 출력시킨다.

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

-case 2

league\_position을 1부터 20사이의 값을 입력하라고 출력하고 입력받기

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

teams.csv에서 awk명령어를 이용하여 leaguePosition변수값과 동일한 라인을 찾아서 해당열의 리그포지션숫자, 팀이름, 승수, 승률을 출력

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

-case 3

Top3 관중을 가진 경기의 데이터를 원하는지 입력해달라고 출력 후 입력받기

=>read -p "Do you want to know Top-3 attendance data and average attendance? (y/n): " check

Check변수가 y인 경우에만 데이터를 출력

=> if [ "$check" = "y" ]

then

fi

matches.csv을 콤마를 구분자로 2번째 열 기준 내림차순 정렬한 후, 상위 3줄만 뽑은 뒤 awk명령어를 이용하여 출력양식에 맞게 출력

=> cat matches.csv | sort -r -t',' -k 2 -g | head -n 3 | awk -F, '{print $3" vs "$4"\("$1"\)\n"$2,$7"\n"}'

-case 4

For문과 seq을 이용하여 1부터 20까지 반복문을 돌린다.

=> For num in $(seq 1 20)

do

done

반복문 안에 파이프와 awk명령어를 이용하여 숫자와 해당 숫자 번째의 팀이름을 출력한다.

=> cat teams.csv | awk -F, -v n=$num '$6==n{print n,$1}'

teams.csv에서 구한 팀이름을 players.csv파일에서 사용하기 위해 변수로 저장한다.

=> team=$(cat teams.csv | awk -F, -v n=$num '$6==n{print $1}')

변수로 들어온 팀에 맞는 선수들 중 골을 가장 많이 넣은 선수를 찾아야하기 때문에 players.csv파일에서 awk명령어로 4번째 필드에서 팀이름변수와 동일한 라인들을 찾아 선수이름과 득점 수를 출력, 그 출력을 입력으로 받아 득점 수를 기준으로 내림차순 정렬을 한 후, 맨 위 한 줄만 출력

=> cat players.csv | awk -F, -v t="$team" '$4==t{print $1","$7}' | sort -r -t',' -k 2 -g | head -n 1

-case 5

수정된 포맷의 날짜를 원하는지 출력하고 입력받기

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

check변수가 y인 경우에만 데이터를 출력

=> if [ "$check" = "y" ]

then

fi

matches.csv에서 위에서부터 11번째까지 받고(첫번째 줄이 설명줄이기 때문이다.) awk명령어를 이용하여 첫번째 줄을 제외시킨다. 그 출력을 입력으로 받아 출력 awk명령어를 한번 더 이용하여 양식에 맞춰 출력한다. 그 출력을 입력받아 sed명령어를 이용하여 Aug를 08로 바꿔 출력한다.

=> cat matches.csv | head -n 11 | awk -F, 'NR!=1{print $1}' | awk '{print $3"/"$1"/"$2,$5}' | sed -E 's/Aug/08/g'

-case 7

Bye!를 출력하고 실행이 종료되어야 하기 때문에 stop을 Y로 바꿔주어 until문이 실행되지 않게 한다.

=> echo "Bye!" stop="Y"

(Github에 push하였는데 default branch가 main으로 되어있는지 모르고 remote를 master로 해서 branch가 두 개가 되었습니다. Main이 아닌 master에 코드파일과 csv이 있습니다.)