**# proj1\_12211640\_ShinEunSeop.sh**

3개의 파일을 받아 7개의 메뉴를 제공하며 7번 메뉴를 선택하기 전까지 무한 반복하는 프로그램이다.

**#사용방법**

./proj1\_12211640\_ShinEunSeop.sh teams.csv players.csv matches.csv

인수가 3개가 아니거나 .csv파일이 아닌 경우 오류 출력 후, 프로그램이 종료된다.

.csv 파일 3개를 넣을 경우

\*\*\*\*\*\*\*\*\*\*\*\*OSS1 - Project1\*\*\*\*\*\*\*\*\*

\* StudentID : 12211640 \*

\* Name : EunSeop Shin \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

이 출력된다.

다만, 파일이 teams.csv players.csv matches.csv 순서가 아닐 경우 원하는 결과가 나오지 않는다.

**# 메뉴1**

손흥민 선수의 데이터를 얻을 수 있다. y 입력 후 정상적으로 이용 가능하고 n을 입력했을 때에는 돌아가 다시 전체 메뉴를 출력한다. 만약 입력이 y와 n이 아닌 경우 오류를 출력한다.

Do you want to get the Heung-Min Son's data? (y/n): y2

ERROR:INVALID INPUT y2

**# 메뉴2**

리그 순위에 따른 팀 이름, 승률 데이터를 보여준다. 1부터 20까지 숫자를 입력해야 하며 문자열이나 범위에 벗어난 숫자 입력 시 메뉴1과 동일한 양식의 오류를 출력한다.

**# 메뉴3**

관객 수 Top-3 매치에 관한 데이터를 보여준다. 이용 방법은 메뉴1과 같다.

**# 메뉴4**

각 팀에서 득점왕인 선수를 리그 팀 리그 순위대로 보여준다. 이용 방법은 메뉴1과 같다.

## 코드 분석

target\_team=$(cat $1 | tail -n +2 | sort -n -t, -k6 | awk -F, '{printf("%s,",$1)}')

target\_team에 각 팀의 이름을 쉼표로 끊어 저장한다.

이유는 Manchester City와 같이 팀 이름에 공백이 포함되어 있는 경우

for team\_ in $target\_team 구문에서 team\_의 값 통제가 불가능하기 때문이다.

쉼표를 구분자로 사용하기 위해 IFS=”,” 사용한다.

cat $2 | tail -n +2 | awk -F, -v team=$team\_ '$4~team {print $1","$7}' | sort -r -n -t, -k2 | tr ',' ' '| head -1 ; echo "";

또한 쉼표로 선수이름과 득점수를 구별하며 득점 수에 따라 정렬 이후 tr로 쉼표를 제거한다.

**# 메뉴5**

매치 정보가 들어있는 파일에 들어있는 날씨 정보를 변환해서 보여준다. 이용 방법은 메뉴1과 같다.

## 코드 분석

cat $3 | sed -ne '2,11p' | sed -e 's/Jan/01/' -e 's/Feb/02/' -e 's/Mar/03/' -e 's/Apr/04/' -e 's/May/05/' -e 's/Jun/06/' -e 's/Jul/07/' -e 's/Aug/08/' -e 's/Sep/09/' -e 's/Oct/10/' -e 's/Nov/11/' -e 's/Dev/12/' | awk '{printf("%d/%s/%d %s\n",$3,$1,$2,$5)}' | cut -d, -f1

sed를 이용하여 각 영문으로 된 월을 양식에 맞게 숫자로 변경한다. 01,02,…,09의 경우 printf에서 %d로 출력하려 하면 0이 사라지므로 %s 문자열로 출력한다.

**# 메뉴6**

홈에서 치른 경기 중 가장 득점 차가 높은 승리 경기에 대한 정보를 보여준다. 각 팀은 숫자로 선택할 수 있다. 이용 방법은 메뉴2와 동일하다.

## 코드 분석

cat $3 | tail -n +2 | awk -F, -v team=$team\_name '$3~team {printf("%s/%s %d vs %d %s/%d\n",$1,$3,$5,$6,$4,$5-$6)}'| sort -t/ -n -r -k3 | awk -F/ 'BEGIN {max=-10000} $3 >= max {printf("%s\n%s\n\n",$1,$2);max=$3}'

(홈 팀 득점 수-원정 팀 득점 수)를 정렬 지표로 삼기 위해 슬래시(/)로 데이터를 구분한다. 이때 마지막 결과 출력 시 날씨 데이터 이후 뉴라인이 들어가야 하기에 날씨 데이터 부분도 슬래시로 구분한다. 정렬 이후 awk BEGIN을 이용하여 max를 선언 및 초기화, 첫 번째 줄의 득점 차 값이 가장 크기 때문에 득점 차($3)>=max를 이용하여 필요한 데이터만 가져온 후 양식대로 출력한다.

**# 메뉴7**

프로그램을 종료한다.