**Opensource SW**

Project 1

12191620 심찬희

1번:

* read -p명령어를 통해 y/n을 입력받습니다(y 입력받으면 진행)
* awk -F ‘,’명령어를 통해 입력 레코드를 쉼표로 구분하도록 설정합니다.
* players.csv에서 $1 첫번째 필드(이름)이 “Heung-Min Son”인 경우 데이터를 출력하도록 한다

2번:

* read -p명령어를 원하는 팀의 순위를 입력받아 position에 넣어둡니다.
* total\_matches = 승+무+패 경기수, win\_rate = 승/total\_matches로 계산하여 승률을 계산합니다.
* awk -F ‘,’ 명령어를 통해 입력 레코드를 쉼표로 구분하도록 설정합니다
* teams.csv에서 순위가 입력받은 position과 같은 경우 팀의 순위, 팀 이름, 승률을 출력합니다

3번

* read -p명령어를 통해 y/n을 입력받습니다(y 입력받으면 진행)
* sort -t’,’ -k2,2nr 명령어를 통해 matches.csv에서 구분자를 ,로 2번째 필드인 attendance를 기준으로 오름차순 정렬합니다
* head -n 3 명령어를 통해 그 중 상위 3개의 데이터를 가져옵니다
* awk -F ‘,’ 명령어를 통해 입력 레코드를 쉼표로 구분하도록 설정합니다
* home/away\_team\_name, home/away\_team\_score, datetime, attendance, stadium 정보를 출력합니다

4번

* read -p명령어를 통해 y/n을 입력받습니다(y 입력받으면 진행)
* awk -F ‘,’명령어를 통해 입력 레코드를 쉼표로 구분하도록 설정합니다
* NR>1으로 의도하지 않은 헤더 데이터를 건너뜁니다.
* team배열에 position을 key로 common\_name을 value로 저장합니다.
* For 루프를 통해 각 key에 대해 순회합니다.
* team\_name이 순위에 해당하는 팀 이름을 가져옵니다
* while 루프는 players.csv의 각 레코드를 가져옵니다
* if 조건문을 통해 선수의 소속팀이 현재 순회중인 팀과 같고 현재까지의 top\_scoerer\_goal보다 더 많은 골을 기록했다면 top\_scorer의 기록을 업데이트합니다
* 결과를 출력합니다

5번

* read -p명령어를 통해 y/n을 입력받습니다(y 입력받으면 진행)
* sed -E 명령어를 통해 월의 이름을 해당하는 숫자로 바꿉니다
* awk 명령어를 사용해 각 줄의 날짜를 처리합니다
* -F ‘ – ‘옵션은 레코드를 “-“로 구분하여 필드를 추출합니다(시간 추출)
* Split()을 통해 첫 번쨰 필드를 공백으로 분할해 ‘a’배열에 저장합니다
* 그리고 날짜를 년/월/일 시간 형식으로 조합하여 출력합니다.
* 시간 필드에 뒤에 다른 필드의 데이터도 같이 붙어 있기 때문에 awk -F ‘,’ {print $1} 명령어로 쉼표로 구분된 필드에서 첫 번째 필드(시간)만 출력합니다
* head -n 10 명령어를 통해 상위 10줄만 출력합니다

6번 find\_largest\_difference\_team\_menu()

* awk -F ‘,’ 명령어를 사용해 필드 구분자를 쉼표로하여 teams.csv의 데이터를 가져옵니다
* NR > 1옵션을 통해 의도하지 않은 헤더 데이터를 건너뜁니다.
* 그리고 각 팀을 순회하며 첫번째 필드(팀 이름)을 출력합니다
* read -p 명령어를 통해 사용자의 입력을 team\_number에 받습니다
* team\_number에 해당하는 팀의 이름을 찾기 위해 -v team\_num=”$team\_number”옵션을 사용해서 사용자가 입력한 team\_number를 전달하고 NR == team\_num + 1 조건을 사용하여 해당 팀의 이름을 가져옵니다.
* 여기서 team\_num가 1부터 시작하는 반면 NR은 0부터 시작하므로 +1 해줍니다
* find\_largest\_difference 함수에 팀의 이름과 번호를 전달합니다.

find\_largest\_difference()

* team\_name과 team\_number를 매개변수로 받습니다.
* matches.csv 파일에서 해당 팀의 홈 경기 결과를 가져옵니다
* 이를 위해 awk 명령어를 사용하여 특정 팀이 홈 팀인 경기만 추출합니다.
* 가장 큰 득점 차이를 저장하기 위한 변수 largest\_difference와 해당 경기의 정보를 저장하기 위한 배열 winning\_matches를 초기화합니다.
* 각 홈 경기에 대해 반복문을 실행합니다. 각 경기에서 홈 팀의 득점과 어웨이 팀의 득점을 비교하여 득점 차이를 계산합니다. 현재 경기의 득점 차이가 기존의 가장 큰 득점 차이보다 큰지 확인합니다.
* 만약 더 크다면, largest\_difference를 업데이트하고 winning\_matches 배열에 해당 경기의 정보를 덮어 씌웁니다(원래 있던 정보는 삭제)
* 만약 현재 경기의 득점 차이가 기존의 가장 큰 득점 차이와 같다면, 해당 경기의 정보를 winning\_matches 배열에 추가합니다.(원래 있던 정보 뒤에 추가)
* winning\_matches 배열에 저장된 경기 정보를 출력합니다