

Opensource SW  
Project 1

12204948 최승혁

```
28 export LC_ALL=C.UTF-8
29 if [ $# -ne 3 ]; then
30     echo "usage: ./prj1_12204948_Choiseunghyuk.sh file1 file2 file3"
31     exit 1
32 fi
```

사용법 오류처리

```
38 echo "*****OSS1 - Project1*****"
39 echo "*           StudentID : 12204948           *"
40 echo "*           Name : Seunghyuk Choi           *"
41 echo "*****"
42
```

who am i

```
# 메인 메뉴
function main_menu {
    echo "[MENU]"
    echo "1. Get the data of Heung-Min Son's Current Club, Appearances, 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"
}
```

메인 메뉴

```
55 function get_player_data() {
56     #players.csv 파일에서 matching player data를 추출합니다
57     local player_name="$1"
58     local player_data=$(grep "$player_name" players.csv)
59
60     if [ -z "$player_data" ]; then
61         echo "Player '$player_name' not found in the data."
62         return 1
63     fi
64     # 팀명, 경기수, 득점, 도움 등을 출력합니다.
65     local full_name=$(echo "$player_data" | cut -d',' -f1)
66     local current_club=$(echo "$player_data" | cut -d',' -f4)
67     local appearances=$(echo "$player_data" | cut -d',' -f6)
68     local goals=$(echo "$player_data" | cut -d',' -f7)
69     local assists=$(echo "$player_data" | cut -d',' -f8)
70
71     echo "Team:$current_club,Appearance:$appearances,Goal:$goals,Assist:$assists\n"
72 }
```

1번 문제

```

74 ~ function get_team_data() {
75     local league_position="$1"
76     # awk를 사용하여 리그 위치를 기반으로 teams.csv 파일에서 팀 데이터를 검색합니다.
77     local team_data=$(awk -F',' ' -v pos="$league_position" '$6 == pos {print $0}' teams.csv)
78
79 ~     if [ -z "$team_data" ]; then
80         echo "No team found for league position $league_position."
81         return 1
82     fi
83
84     local team_name=$(echo "$team_data" | cut -d',' -f1)
85     local wins=$(echo "$team_data" | cut -d',' -f2)
86     local draws=$(echo "$team_data" | cut -d',' -f3)
87     local losses=$(echo "$team_data" | cut -d',' -f4)
88
89     # 총 게임 수를 계산합니다 (승리 + 무승부 + 패배).
90     local total_games=$((wins + draws + losses))
91     # 승률을 계산합니다.
92     local winning_rate=$(awk "BEGIN {printf \"%.6f\\\", $wins / $total_games}")
93
94     # 리그 위치, 팀 이름, 승률을 출력합니다.
95     echo "$league_position $team_name $winning_rate\\n"
96 }

```

## 2번 문제

```

99 ~ function get_top_attendance() {
100     # matches.csv 파일을 출석률(2번째 열)을 기준으로 내림차순으로 정렬합니다.
101     local sorted_matches=$(sort -t',' -k2 -nr matches.csv)
102     # 정렬된 경기 중 상위 3개를 선택합니다.
103     local top_matches=$(echo "$sorted_matches" | head -n 3)
104     echo "***Top-3 Attendance Match***\\n"
105
106     # 각 경기의 정보를 읽어와서 홈 팀, 원정 팀, 날짜, 출석률, 경기장을 출력합니다.
107 ~     echo "$top_matches" | while IFS=',' read -r date attendance home_team away_team stadium; do
108         echo "$home_team vs $away_team ($date)"
109         echo "$attendance $stadium"
110         echo ""
111     done
112 }

```

## 3번 문제

```

114 # 이 함수는 각 팀의 순위와 최고 득점자를 가져옵니다.
115 ~ function get_team_ranking_and_top_scorer() {
116     read -p "Do you want to get each team's ranking and the highest-scoring player? (y/n) : " answer
117 ~     if [[ $answer != "y" ]]; then
118         return
119     fi
120
121     IFS=$'\n'
122
123 ~     for team_data in $(awk -F',' '{print $6 "," $1}' teams.csv | sort -n); do
124         IFS=,
125         # 팀 순위와 팀 이름을 추출합니다.
126         team_rank=$(echo "$team_data" | cut -d ' ' -f1)
127         team_name=$(echo "$team_data" | cut -d ' ' -f2)
128
129         # players.csv 파일에서 해당 팀의 최고 득점자를 찾습니다.
130         top_scorer=$(awk -F ' ' ' -v team="$team_name" '$4 == team {print $1 " " $7}' players.csv | sort -nr -k 3 | head -n 1)
131
132         # 팀 순위, 팀 이름, 최고 득점자를 출력합니다.
133         printf "%d %s\\n" "$team_rank" "$team_name"
134         printf "%s\\n\\n" "$top_scorer"
135     done
136     IFS=' '
137 }
138

```

## 4번 문제

```

138
139 # 시간의 포맷을 바꿉니다.
140 function format_date_gmt() {
141     counter=0
142
143     # matches.csv 파일에서 날짜를 추출하고, 상위 10개의 날짜에 대해 반복합니다.
144     awk -F ',' '{if(NR!=1) print $1}' matches.csv | head -n 10 | while read -r date_string; do
145         if ((counter <= 9)); then
146
147             # 날짜를 공백으로 분리하여 배열에 저장합니다.
148             IFS=' ' read -r -a date_array <<< "$date_string"
149
150             # 월, 일, 년, 시간을 추출합니다.
151             month="${date_array[0]}"
152             day="${date_array[1]}"
153             year="${date_array[2]}"
154             time="${date_array[4]}"
155
156             # 월의 약어를 숫자로 변환합니다.
157             month=$(sed 's/Aug/08/g' <<< "$month")
158
159             # 포맷된 날짜를 출력합니다.
160             printf "%s/%s/%s %s\n" "$year" "$month" "$day" "$time"
161
162             ((counter++))
163         fi
164     done
165
166     # 모든 포맷된 날짜를 출력합니다.
167     for ((i=0; i<${#date_formatted[@]}; i++)); do
168         echo "${date_formatted[$i]}"
169     done
170     echo ""
171 }
172

```

5번 문

제

```

173 # 팀들을 모두 출력합니다.
174 display_team_choices() {
175     awk -F ',' '{if (NR!=1) print NR-1 " " " $1}' teams.csv
176 }
177
178 # 팀이름을 가져옵니다.
179 get_team_name() {
180     local team_number="$1"
181     local team_name=$(awk -F ',' "NR==$team_number+1 { print \$0 }" teams.csv | awk -F ',' '{ print $1 }')
182     echo "$team_name"
183 }
184

```

6번 문제 - 1

```

185 # 홈 경기에서 가장 큰 차이로 이긴 팀을 찾습니다.
186 function find_largest_home_win() {
187     local team_name=$(get_team_name "$1")
188     echo "Team: $team_name"
189     largest_diff=0
190     best_matches=""
191
192     # matches.csv 파일에서 각 경기를 읽어와서 홈 팀이 주어진 팀인 경우에 대해 처리합니다.
193     while IFS=, read -r date_GMT attendance home_team_name away_team_name home_team_goal_count away_team_goal_count stadium_name; do
194         if [[ "$home_team_name" == "$team_name" ]]; then
195             # 홈 팀과 원정 팀의 골 수를 가져옵니다.
196             home_goals=$(echo "$home_team_goal_count" | tr -d ' ')
197             away_goals=$(echo "$away_team_goal_count" | tr -d ' ')
198             # 골 차이를 계산합니다.
199             goal_diff=$((home_goals - away_goals))
200
201             # 날짜를 분리합니다.
202             IFS=' ' read -r -a date_array <<< "$date_GMT"
203             month="${date_array[0]}"
204             day="${date_array[1]}"
205             year="${date_array[2]}"
206             time="${date_array[4]}"
207
208             # 골 차이가 같은 경우, 여러 경기를 저장합니다.
209             if [[ $goal_diff -gt $largest_diff ]]; then
210                 largest_diff=$goal_diff
211                 best_matches="$month $day $year - $time\nhome_team_name $home_team_goal_count vs $away_team_goal_count $away_team_name\n\n"
212             elif [[ $goal_diff -eq $largest_diff ]]; then
213                 best_matches+="$month $day $year - $time\nhome_team_name $home_team_goal_count vs $away_team_goal_count $away_team_name\n\n"
214             fi
215         fi
216     done < matches.csv
217     echo ""
218     echo "$best_matches"
219 }
220

```

## 6번 문제 -2

```
221 # 메인 메뉴를 표시하고 사용자 입력을 처리합니다.
222 while true; do
223     main_menu
224     read -p "Enter your CHOICE (1~7) : " choice
225
226     if [ $choice -eq 1 ]; then
227         read -p "Do you want to get the Heung-Min Son's data? (y/n) : " confirm
228
229         if [ "$confirm" == "y" ] || [ "$confirm" == "Y" ]; then
230             get_player_data "Heung-Min Son"
231         else
232             echo "Exiting..."
233         fi
234     elif [ $choice -eq 2 ]; then
235         read -p "What do you want to get the team data of league_position[1~20] : " league_position
236         if [ "$league_position" -ge 1 ] && [ "$league_position" -le 20 ]; then
237             get_team_data "$league_position"
238         else
239             echo "Invalid league position. Please enter a value between 1 and 20."
240         fi
241     elif [ $choice -eq 3 ]; then
242         read -p "Do you want to know Top-3 attendance data and average attendance? (y/n) : " choice
243
244         if [ "$choice" == "y" ] || [ "$choice" == "Y" ]; then
245             get_top_attendance
246         else
247             echo "Exiting..."
248         fi
249     elif [ $choice -eq 4 ]; then
250         get_team_ranking_and_top_scorer
251     elif [ $choice -eq 5 ]; then
252         format_date_gmt
253     elif [ $choice -eq 6 ]; then
254         display_team_choices
255         read -p "Enter team number: " team_number
256         find_largest_home_win "$team_number"
257     elif [ $choice -eq 7 ]; then
258         echo "Bye!"
259         exit 0
260     else
261         echo "Invalid option. Please try again."
262     fi
263 done
```

메인 무한 루프