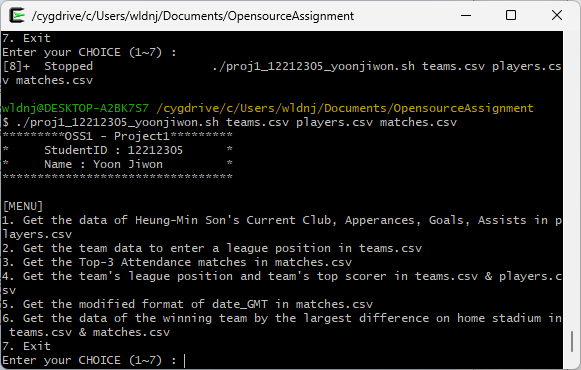
**오픈소스SW개론 – Project 1**

12212305 윤지원

**<실행 방법>**



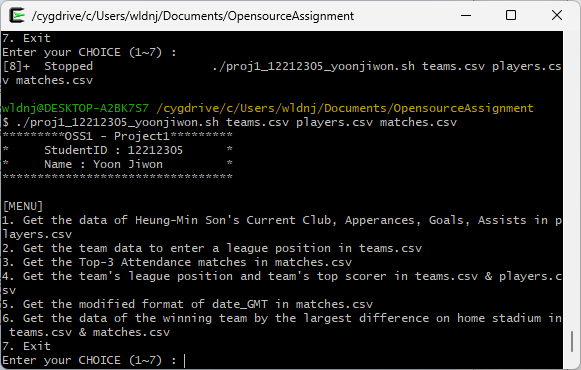
**다음의 경우 오류 발생**

1. 3개의 파일을 함께 사용하지 않는 경우  
   텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

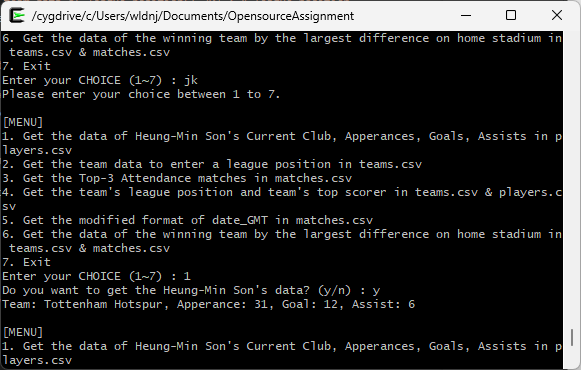
   자동 생성된 설명
2. 입력된 파일의 확장자가 적절하지 않은 경우  
   텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

   자동 생성된 설명

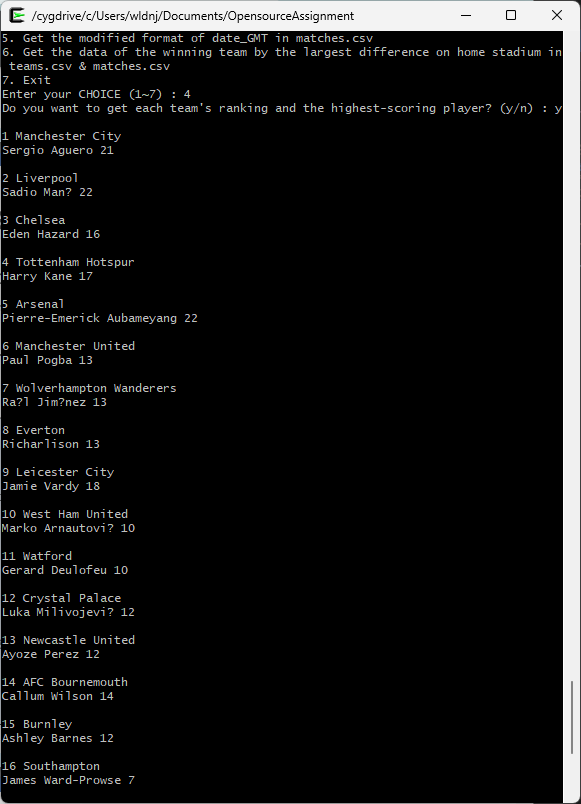
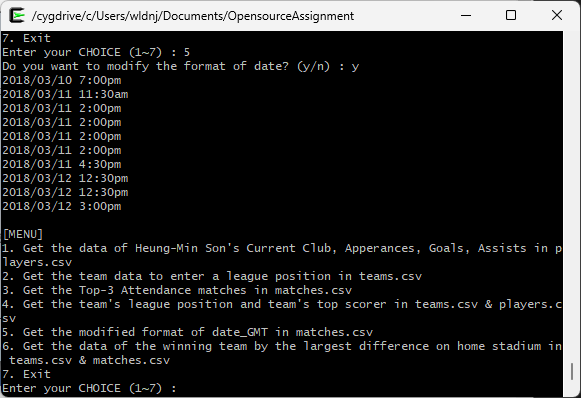
**<기능>**



1 ~ 7번까지 원하는 기능의 번호를 입력하여 실행 가능

1. 손흥민의 현재 팀, 출전 횟수, 골 수, 어시스트 수를 출력  
   
2. 입력한 리그 순위에 해당하는 팀의 정보 출력  
   텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

   자동 생성된 설명
3. 관중 수가 가장 많았던 상위 3개의 경기 출력  
   텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

   자동 생성된 설명
4. 팀의 리그 순위와 각 팀의 최고득점자 출력(최고득점자가 여러 명일 경우, 한 명만 출력)  
   
5. 경기 날짜 형식을 변경하여 출력  
   
6. 입력한 팀이 홈 경기장에서 최다 득실차로 승리한 경기의 정보를 모두 출력  
   텍스트, 전자제품, 스크린샷, 소프트웨어이(가) 표시된 사진

   자동 생성된 설명

**<구현 세부 사항>**

1. 스크립트 실행 시, 입력 확인

if [ $# -lt 3 ]; then

    echo "usage: ./2024-OSS-Project1.sh file1 file2 file3"

    exit 1

elif [[ $1 != \*.csv ]] || [[ $2 != \*.csv ]] || [[ $3 != \*.csv ]]; then

    echo "error: Invalid file format."

    exit 1

fi

스크립트를 실행하면서 올바른 파일을 함께 사용하는지 검사한다.

1. Menu 구현

answer=0

while [ $answer != 7 ]

do

    echo -e

    echo "[MENU]"

    . . . # 메뉴 선택지 출력

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

    case "$answer" in

    . . . # 1~6번 case 실행 부분 구현

    7)

        echo "Bye!" ;;

    \*)

        echo "Please enter your choice between 1 to 7." ;;

    esac

done

while문을 이용하여 사용자가 종료(7)을 입력할 때까지 반복하고, 종료(7)이 입력되면 loop를 종료한다. Case 문을 이용하여 사용자의 입력에 따라 적절한 기능을 수행한다.

1. 기능 구현

4)

- team\_ranking은 각 팀 이름을 league\_position을 기준으로 오름차순 정렬한 data를 출력한다.  
- team\_ranking에서 각 팀의 이름은 개행 문자로 구분되어 있으므로 for문을 돌 때 개행 문자를 기준으로 팀 이름을 가져올 수 있게 IFS를 %’\n’로 설정하였다.   
- Data를 출력할 때, 파일 최상단의 속성 이름값은 출력하지 않도록 NR을 이용해 조건을 설정하였다.

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

        if [ $yORn = y ]; then

            team\_ranking=$(cat teams.csv | sort -n -t, -k6 | cut -d, -f1 | awk 'NR>1')

            rank=1

            IFS=$'\n'

            for var in $team\_ranking;

            do

                echo -e

                echo $rank $var

                cat players.csv | grep "$var" -a | sort -rn -t, -k7 | awk -F ',' 'NR==1 {printf("%s %d\n",$1, $7)}'

                ((rank++))

            done

        fi ;;

6)

- 모든 팀의 이름을 추출하고 select 문을 이용해 정보를 얻고 싶은 팀의 번호를 입력하도록 구현하였다.  
- max\_difference\_score는 for문에서 최다득실차가 몇 점인지를 저장한다. 이후, 저장된 최다득실차와 같은 값을 갖는 data들을 모두 출력한다.

- home\_team\_won\_matches는 입력 받은 팀이 홈 경기장에서 승리한 모든 경기를 출력한다.

IFS=$'\n'

        PS3="Enter your team number : "

        select var in $(cat teams.csv | cut -d, -f1 | awk 'NR>1')

        do

            max\_difference\_score=0

            home\_team\_won\_matches=$(cat matches.csv | awk -F ',' -v team=$var '$3==team && $5>$6')

            for var2 in $home\_team\_won\_matches;

            do

                home\_score=$(echo $var2 | cut -d, -f5)

                away\_score=$(echo $var2 | cut -d, -f6)

                cur\_score=$(($home\_score-$away\_score))

                if [ $max\_difference\_score -lt $cur\_score ]; then

                    max\_difference\_score=$cur\_score

                fi

            done

            cat matches.csv | awk -F ',' -v team=$var '$3==team && '"$max\_difference\_score"'==($5-$6) {printf("\n%s\n%s %d vs %d %s\n", $1, $3, $5, $6, $4)}'

            break

        done

        ;;

*(구현이 짧고 간단한 1, 2, 3, 5번에 대한 세부 설명은 생략)*