12191633 윤호준

./prj1\_12191633\_yunhojun

으로 프로그램을 실행합니다.

\*같은 디렉토리 안에 u.data u.user u.item 파일이 있어야 합니다.\*

1. 프로그램을 실행하면 제 이름과 학번이 출력되고 각 기능에 대한 설명이 출력됩니다.  
   그리고 1에서 9사이의 수를 입력하면 입력에 따른 기능이 실행됩니다. 숫자가 아니거나 1에서 9사이가 아닌 수를 입력하면 아무것도 실행되지 않고 다시 입력을 받습니다. 각 기능이 종료된 후 다시 입력을 받는 곳으로 돌아옵니다.  
     
   ./prj1\_12191633\_yunhojun으로 프로그램을 실행하면 먼저 Manual함수를 호출합니다.  
   Manual함수는 제 이름과 학번, 각 기능에 대한 설명을 출력합니다. 다음으로 숫자를 입력 받아 input변수에 저장하고 case/esac 문을 통해 그 값에 따른 함수를 실행합니다. 이 과정은 while문으로 반복됩니다.
2. 1을 입력하면 movie id를 입력 받았을 때 해당 영화에 대한 정보를 u.item에서 받아서 출력합니다. 1에서 1682사이의 수가 아닌 것을 입력하면 아무것도 출력되지 않습니다.  
     
   먼저 사용자로부터 id를 입력 받습니다. Cat 명령어를 통해 u.item의 데이터를 awk 명령어로 전달합니다. awk내부에서는 입력 받은 id를 변수 a로 사용합니다. 또한 각 필드는 ‘|’에 의해 구분됩니다. 데이터의 각 행에 대하여 첫번째 필드가 a(id)와 같다면 그 행을 출력합니다.  
   cat u.item | awk -v a=$id -F\| '$1 == a {print}'
3. 2를 입력한 뒤 y를 입력하면 action장르에 해당하는 영화를 u.item에서 찾아 첫 10개까지 출력합니다. n을 입력하면 아무것도 출력되지 않습니다.  
     
   먼저 사용자로부터 y혹은 n을 입력 받습니다. y를 입력 받았다면 cat명령어를 통해 u.item의 데이터를 awk명령어로 전달합니다. 데이터의 각 필드는 ‘|’에 의해 구분됩니다. 데이터의 각 행에 대하여 7번째 필드(action)이 1인 경우 첫번째 필드(movie id)와 두번째 필드(title)이 출력됩니다. 이 출력은 다시 두번째 awk로 전달되어 첫 10개의 행만 출력되도록 합니다.  
   cat u.item | awk -F\| '$7 == 1 {print $1, $2}' | awk 'NR<=10 {print}'
4. 3을 입력하면 다음으로 입력된 movie id에 해당하는 영화의 평균 평점을 출력합니다. 1에서 1682사이의 수가 아닌 것을 입력하면 -nan이 출력될 수 있습니다.  
     
   먼저 사용자로부터 movie id를 입력 받습니다. 그리고 cat명령어를 통해 u.data에서 데이터를 가져와 awk명령어로 전달합니다. awk내부에서는 유저가 입력한 movie id를 변수a로 사용합니다. 데이터의 각 행에 대하여 두번째 필드(movie id)가 a(movie id)와 같다면 변수 b에 세번째 필드(rating)을 더하고 변수 c에 1을 더합니다. 이러면 b는 모든 rating의 합, c는 평가한 사람의 수가 됩니다. END를 통해 마지막에 평균(b/c)를 출력합니다.  
   cat u.data | awk -v a=$mid '$2 == a {b+=$3; c+=1} END {print b/c}'
5. 4를 입력한 뒤 y를 입력하면 u.item에서 첫 10개의 영화에 대해 ‘IMDb URL’을 제거하여 출력합니다. n을 입력하면 아무것도 출력하지 않습니다.  
     
   먼저 사용자로부터 y혹은 n을 입력 받습니다. y를 입력 받았다면 cat명령어를 통해 u.item의 데이터를 가져와 awk명령어로 전달합니다. awk에서는 데이터의 첫 10개의 행만 sed명령어로 전달합니다. sed명령어에서는 s명령어를 사용하여 ‘http’로 시작하고 ‘|’로 끝나는 부분을 찾아 ‘|’로 변환하여 IMDb URL을 제거합니다.  
   cat u.item | awk -F\| 'NR<=10 {print}' | sed 's/http[^|]\*|/|/g'
6. 5를 입력한 뒤 y를 입력하면 u.user에서 첫 10명의 사람에 대한 정보를 출력합니다.  
   출력 양식은 “user {name} is {age} years old {gender} {occupation}” 입니다. n을 입력하면 아무것도 출력하지 않습니다.  
     
   먼저 사용자로부터 y혹은 n을 입력 받습니다. Y를 입력 받았다면 cat명령어를 통해 u.user의 데이터를 awk명령어로 전달합니다. awk명령어는 데이터의 첫 10개 행을 sed명령어로 전달합니다. 첫 sed명령어는 s명령어를 통해 ‘M’을 ‘male’로 변환합니다. 두번째 sed명령어는 s명령어를 통해 ‘F’를 ‘female’로 변환합니다. 이렇게 성별양식을 변환한 데이터는 두 번째 awk명령어를 통해 전달되고 ‘|’에 의해 필드를 구분하여 출력 양식에 맞게 출력합니다. 출력 양식은 “user {($1)name} is {($2)age} years old {($3)gender} {($4)occupation}”입니다  
   cat u.user | awk -F\| 'NR<=10' | sed 's/M/male/' | sed 's/F/female/' | awk -F\| '{print "user " $1 " is " $2 " years old " $3 " " $4}'
7. 6을 입력하면 u.item에서 마지막 10개의 영화에 대해 날짜 형식만 변환하여 출력합니다.  
   출력 양식은 “YYYYMMDD”으로 변환됩니다.  
     
   cat 명령어를 통해 u.item의 데이터를 가져와 awk명령로 전달합니다. awk에서는 각 필드를 ‘-’로 구분하고 BEGIN에서 months배열에 각 달의 영문형식을 저장합니다. 그리고 마지막 10개의 행에 대하여 두번째 필드(month)를 months배열에서 찾아 숫자형식으로 변환하여 출력합니다. 이 출력은 sed명령어를 통해 “DD-MM-YYYY” 형식에서 “YYYYMMDD”형식으로 변환됩니다.  
   cat u.item | awk -F \- 'BEGIN {  
    split("Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec", months, " ")  
   }  
    NR >= 1673{  
    for(i = 1; i <= 12; i++){  
    if(months[i] == $2) {  
    $2 = sprintf("%02d", i)  
    break  
    }  
    }  
    print  
   }' | sed -E 's/([0-9]{2}) ([0-9]{2}) ([0-9]{4})/\3\2\1/g'
8. 7을 입력하면 user id를 입력 받았을 때, 해당 유저가 평가한 영화들의 movie id를 오름차순으로 정렬하여 출력한 뒤, 그 중 첫 10개의 영화에 대해 movie id와 제목을 출력합니다  
     
   먼저 user id를 입력 받습니다. 그리고 cat명령을 통해 u.data의 데이터를 가져와 awk명령어로 전달합니다. awk에서는 입력 받은 user id를 변수 a로 사용합니다. 각 행에 대하여 첫번째 필드(user id)가 변수 a(user id)와 같다면 두번째 필드(movie id)를 정렬하여 출력하고, 이는 out변수에 저장됩니다. 출력은 sed명령어를 통해 공백을 ‘|’로 변환한 뒤 이뤄집니다.  
   out=$(cat u.data | awk -v a=$uid '$1 == a{print $2}' | sort -n)  
   echo $out | sed 's/\s/|/g'  
   out 변수를 배열로 사용하기 위해 배열 arr로 변환합니다. 그리고 cat 명령어를 통해 u.item의 데이터를 가져와 awk로 전달합니다. awk에서는 배열 arr의 i번째 원소를 변수a로 사용하며 데이터의 각 필드는 ‘|’로 구분됩니다. 데이터의 각 행에 대하여 첫번째 필드(movie id)가 a(arr[i])와 같다면, 첫번째 필드(movie id)와 두번째 필드(title)을 출력합니다. 이 과정은 for문을 통해 반복되며 i는 for문의 iterator입니다.  
   arr=($out)  
   for i in $(seq 0 9)  
   do  
    cat u.item | awk -v a="${arr[$i]}" -F \| '$1==a{print $1 "|" $2}'  
   done
9. 20대 프로그래머들이 평가한 모든 영화들에 대하여 해당 영화의 movie id와 20대 프로그래머들의 평균 평점을 출력합니다. 만약 평가하지 않은 영화가 있다면 출력되지 않습니다. 이 과정은 수 분이 걸릴 수 있습니다.  
     
   먼저 사용자로부터 y혹은 n을 입력 받습니다. y를 입력 받았다면 cat명령어를 통해 u.user의 데이터를 가져와 awk명령어로 전달합니다. awk에서는 각 필드를 ‘|’로 구분합니다. 각 행에 대하여 두번째 필드(age)가 20이상이고 30미만이면서 4번째 필드(occupation)이 ‘programmer’라면 첫번째 필드(user id)를 uid변수에 저장합니다.  
   uid=$(cat u.user | awk -F \| '$2>=20 && $2<30 && $4=="programmer"{printf "%s ", $1}')  
   다음으로 cat명령어를 통해 u.data의 데이터를 가져와 awk명령어로 전달합니다. awk에서는 uid를 변수a로 i를 변수 mid로 사용합니다. BEGIN에서는 변수a(uid)를 공백으로 구분하여 배열 arr에 저장합니다. 데이터의 각 행에 대하여 두번째 필드(movie id)가 mid(i)와 같다면 for문을 통하여 첫번째 필드(user id)가 배열에 존재하는지 확인합니다. 만약 존재한다면 b에 세번째 필드(rating), c에 1을 더합니다. 이러면 b는 해당 영화의 평점의 합, c는 평가한 사람의 수가 되고 END에서는 b가 0이 아니라면 평균(b/c)을 출력합니다. 이 과정은 for문을 통해 모든 영화에 대하 반복합니다.