

---

```
#!/bin/bash
```

```
item=$1
data=$2
user=$3
```

```
input=0
```

맨 위의 "#! /bin/bash"는 Shebang 으로, 어떤 경로에서든 동일하게 실행할 수 있도록 했습니다.

아래의 세 변수, item, data, user 는 각각 u.item, u.data, u.user 파일을 입력받기 위한 변수입니다.

Input 은 MENU를 선택하는 변수입니다. 0으로 초기화 시켜두었습니다.

```
echo "-----"
echo "User Name: Yeo Inho"
echo "Student Number: 12201759"
echo "[MENU]"
echo "1. Get the data of the movie identified by a specific 'movie id' from 'u.item'"
echo "2. Get the data of action genre movies from 'u.item'"
echo "3. Get the average 'rating' of the movie identified by specific 'movie id' from 'u.data'"
echo "4. Delete the 'IMDb URL' from 'u.item'"
echo "5. Get the data about users from 'u.user'"
echo "6. Modify the format of 'release date' in 'u.item'"
echo "7. Get the data of movies rated by a specific 'user id' from 'u.data'"
echo "8. Get the average 'rating' of movies rated by users with 'age' between 20 and 29 and 'occupation' as 'programmer'"
echo "9. Exit"
echo "-----"
```

출력하는 부분입니다. 맨 위에는 제 이름, 학번이 있고 과제에서 주어진 것 처럼 처음 한 번만 출력했습니다.

```
while true;
do
    read -p "Enter your choice [ 1-9 ] " input
```

while true; 로 루프를 돌게 만들었습니다. 아래에 input 이 9 일때 break구문이 실행하여 루프를 빠져나갈 수 있게 만들었습니다. 밑에서 다시 설명하겠습니다.

read의 -p 옵션을 통해 input에 입력을 받기 전에 안내문구를 출력하였습니다.

input 에는 1부터 9까지의 숫자를 입력받습니다.

```
if [ $input -eq 1 ]
then
    read -p "Please enter 'movie id' (1~1682):" mov_id
    cat $item | awk -F| -v _id=$mov_id '_id==$1{print $0}'
```

awk를 사용했고, u.item 은 구분자가 ' ' 가 아닌 '|' 로 되어있기 때문에 -F'|' 옵션을 통해 구분자를 변경하였습니다. 또한 -v \_id=\$mov\_id 를 통해서 외부 변수를 \_id 에 저장하였고 "\_id==\$1{print \$0}" 을 통해서 영화의 번호가 일치하는 행을 찾아 전부 출력하였습니다.

[illegible]

```
elif [ $input -eq 2 ]
then
    read -p "Do you want to get the data of 'action' genre movies from 'u.item'? (y/n):" yn
    if [ $yn != "y" ]
    then continue;
    fi
    cat $item | awk -F\| '{print $1,$2}' | head -n 10
```

주어진 u.item 은 영화번호 오름차순으로 정렬되어 있기 때문에 action장르만 찾아서 상위 10만 뽑으면 됩니다. u.item 에서 7번째 Column 이 1이면 Action 이므로 awk를 통해서 \$1 \$2, 즉 영화 번호와 영화 제목을 전부 뽑고 "head -n 10"을 통해 상위 10개만 추출하였습니다.

```

Enter your choice [ 1-9 ] 2
Do you want to get the data of 'action' genre movies from 'u.item'?(y/n):n
Enter your choice [ 1-9 ] 2
Do you want to get the data of 'action' genre movies from 'u.item'?(y/n):y
2 GoldenEye (1995)
4 Get Shorty (1995)
17 From Dusk Till Dawn (1996)
21 Muppet Treasure Island (1996)
22 Braveheart (1995)
24 Rumble in the Bronx (1995)
27 Bad Boys (1995)
28 Apollo 13 (1995)
29 Batman Forever (1995)
33 Desperado (1995)
Enter your choice [ 1-9 ]

```

3.

```

elif [ $input -eq 3 ]
then
    read -p "Please enter 'movie id' (1~1682):" mov_id
    cat $data | awk -v _id=$mov_id ' _id==$2{sum+= $3} _id==$2{cnt+=1} END {printf "%.5f\n", sum/cnt}'

```

input이 3일 경우 실행되는 부분입니다.

mov\_id 로 영화의 번호를 받았습니다.

u.data 에서 mov\_id 에 해당하는 영화의 모든 rating의 평균을 구해야 합니다.

awk 내부변수 sum과 cnt를 사용하여 mov\_id가 일치하는 모든 rating의 합과 개수를 구하고 sum/cnt 를 통해 평균을 구했습니다.

소수점 아래 6자리에서 반올림하기 위해서 printf 의 "%.5f" 를 사용했습니다.

```

Enter your choice [ 1-9 ] 3
Please enter 'movie id' (1~1682):1
3.87832
Enter your choice [ 1-9 ] 

```

4.

```

elif [ $input -eq 4 ]
then
    read -p "Do you want to delete the 'IMDb URL' from 'u.item'?(y/n):" yn
    if [ $yn != "y" ]
    then continue;
    fi
    cat $item | head -n 10 | sed -E 's/http:[^)]*)\\)/g'

```

input 이 4일 경우 실행되는 부분입니다.

yn을 받고 y일 경우 아래를 실행합니다.

성별을 구분짓는 M과 F외에는 대문자를 사용하지 않으므로 sed를 이용하여 별다른 조건없이 M

은 male로, F는 female로 바꾸어줍니다.

sed의 -e 옵션을 통해 두 식을 연달아 사용할 수 있었습니다.

```
Enter your choice [ 1-9 ] 5
Do you want to get the data about users from 'u.user'? (y/n):y
user 1 is 24 years old male technician
user 2 is 53 years old female other
user 3 is 23 years old male writer
user 4 is 24 years old male technician
user 5 is 33 years old female other
user 6 is 42 years old male executive
user 7 is 57 years old male administrator
user 8 is 36 years old male administrator
user 9 is 29 years old male student
user 10 is 53 years old male lawyer
Enter your choice [ 1-9 ]
```

## 6.

```
elif [ $input -eq 6 ]
then
    read -p "Do you want to Modify the format of 'release data' in 'u.item'? (y/n):" yn
    if [ $yn != "y" ]
    then continue;
    fi
    cat $item | awk -F| ' $1>=1673&&$1<=1682{print $0}' | sed -Ee 's/([0-9][0-9])(-...)(-[0-9][0-9][0-9][0-9])/
\3\2\1 /g' -e 's/-Jan/01/g' -e 's/-Feb/02/g' -e 's/-Mar/03/g' -e 's/-Sep/09/g' -e 's/-Oct/10/g' -e 's/\\| \-|/g' -e 's/
\\|/g'
```

input 이 6인 경우 실행되는 부분입니다.

일단 awk를 통해서 영화의 번호가 1673~1682 인것만 추출하였습니다.

그 후에 일, 월, 연을 추출합니다.

일 : 0-9 숫자 두개

월 : - 뒤에 ...(문자 세 개)

연 : -뒤에 0-9 숫자 네개

후에 back reference를 통해 위치를 바꿔주었습니다.

하지만, 이대로는 "| -1991-Jan02 |" 와 같은 형식이 됩니다.

그 후에 -e로 연달아 쓴 4개의 식은 "-..." 즉 -후에 알파벳으로 쓰여진 달을 숫자로 바꾸어줍니다.

"| -19910102 |" 와 같이 바뀔 수 있습니다.

그 다음 's/\\| \-|/g' 는 연 앞에 있던 공백과 - 기호를 없애줍니다.

"|19910102 |" 와 같이 바뀝니다.

그 다음 's/ w/|/g' 는 맨 뒤의 공백을 없애줍니다. 따라서 주어진 과제와 똑같이 출력할 수 있습니다.

[illegible]

7.

```

elif [ $#input -eq 7 ]
then
    read -p "Please enter the 'user id' (1~943):" user_id
    tmp=$(cat $data | awk -v _id=$user_id '_id==$1{print $2}' | sort -n)
    echo $tmp | tr ' ' '|'

    tmp=$(echo $tmp | tr ' ' '\n' | head -n 10 | tr '\n' ' ')
    echo ""

    for j in $tmp
    do
        cat $item | awk -F\| -v _comp=$j '_comp==$1{printf("%d|%s\n",$1, $2)}'
    done
done

```

input0이 7인 경우 실행되는 부분입니다.

입력으로 user\_id를 받습니다. 해당 user가 평가한 영화 번호들을 |로 구분하여 오름차순으로 출력하고 상위 10개 영화의 영화번호와 제목을 |와 wn으로 구분하여 출력하여야 합니다.

tmp변수에 data중 user\_id가 일치하는 행의 2번째 요소(평가한 영화의 번호)만 sort -n (숫자 정렬) 하여 대입합니다.

이를 `tr '' ''`를 통해 공백을 전부 |로 바꿔주면 첫번째 출력물을 출력할 수 있습니다.

두번째는 tmp의 공백을 개행문자로 바꾸어주고 상위 10개만 추출합니다.

그 후의 echo "" 는 format 을 맞추기 위한 줄바꿈 입니다.

for j in \$tmp 는 tmp의 모든 요소를 돌며 반복합니다.

tmp에는 영화들의 번호가 들어있으므로 awk를 통해서 u.item 에서 영화의 번호가 일치하는 행을 찾고 그 행에서 첫번째, 두번째 요소를 출력할 수 있습니다.

printf 를 통해서 format을 맞춰주었습니다.

```
Enter your choice [ 1-9 ] 7
Please enter the 'user id' (1~943):12
4|15|28|50|69|71|82|88|96|97|98|127|132|133|143|157|159|161|168|170|172|174|191|
195|196|200|202|203|204|215|216|228|238|242|276|282|300|318|328|381|392|402|416|
471|480|591|684|708|735|753|754

4|Get Shorty (1995)
15|Mr. Holland's Opus (1995)
28|Apollo 13 (1995)
50|Star Wars (1977)
69|Forrest Gump (1994)
71|Lion King, The (1994)
82|Jurassic Park (1993)
88|Sleepless in Seattle (1993)
96|Terminator 2: Judgment Day (1991)
97|Dances with Wolves (1990)
Enter your choice [ 1-9 ]
```

## 8.

```

elif [ $input -eq 8 ]
then
    read -p "Do you want to get the average 'rating' of movies rated by users with 'age' between 20 and 29 and
'occupation' as 'programmer'?(y/n):" yn
    if [ $yn != "y" ]
    then continue;
    fi

    young=$(cat $user | awk -F\| ' $2>=20&&$2<=29&&$4=="programmer"{print $1}')

    for i in $young
    do
        cat $data | awk -v _ind=$i ' $1==_ind{print $2, $3 >> "Inho_tmp.txt"}'
    done

    count=$(cat $item | wc -l)

    comp=1
    until [ $comp -gt $count ]
    do
        cat Inho_tmp.txt | awk -v _ind=$comp ' _ind==$1{sum+=$2} _ind==$1{cnt+=1} END {if(cnt!=0)printf("%d, %.
5f\n", _ind, sum/cnt)}' | sed -Ee 's/0+$/g' -e 's/\.$/g'
        comp=$(( $comp+1 ))
    done

```

input 이 8인 경우 실행되는 부분입니다.

일단 young 에 20대 프로그래머들의 유저아이디를 ' '로 구분하여 저장합니다.

young 에 대해 for 문을 돌려서 "Inho\_tmp.txt" 에 Redirection을 통해서(1은 생략) 20대 프로그래머들이 평가한 영화코드와 rating을 저장합니다. (약 30000개)

count는 모든 영화의 개수입니다.

따라서, count 만큼 반복문을 돌립니다. 이때, i와 영화코드가 일치하면 sum과 cnt를 바꾸어주고 END 를 통해 연산이 전부 끝난 후 cnt 가 0이 아닌 경우에만 .5f 로 소숫점 6자리에서 반올림을 해줍니다.

이때, "4.00000"과 같이 불필요한 0과 소숫점을 없애기 위해 sed를 사용했습니다.

0+\$ 를 통해 오른쪽끝에서부터 연속되는 0을 전부 지우고, -e 옵션을 통해 w.\$ 로 오른쪽 끝에있는 소숫점들을 전부 지웠습니다.



```
Enter your choice [ 1-9 ] 8
Do you want to get the average 'rating' of movies rated by users with 'age' betw
een 20 and 29 and 'occupation' as 'programmer'?(y/n):y
1, 4.29412
2, 3
3, 3.5
4, 3.7
5, 3.25
7, 4.22222
8, 3.5
9, 4.1
10, 4
11, 4.3125
12, 4.69231
13, 3.375
14, 4
15, 3.85714
16, 3
```

9.

```
elif [ $input -eq 9 ]
then
    echo "Bye!"
    break;

fi

done
```

input 이 9일 경우 실행되는 부분입니다.

Bye! 를 출력하고 break로 처음 설명했던 while true; 를 빠져나옵니다.

아래의 fi는 맨 처음 if [ \$input -eq 1 ] 과 대응되는 부분이며

done 또한 break 로 빠져나가는 while의 do와 대응되는 부분입니다.

따라서 input 에 9를 입력하면 프로그램이 종료됩니다.