Bash Shell Programming

OSS Project 1

제출일	2023.11.02	전공	인공지능공학과
과목	오픈소스SW개론	학번	12221828
담당교수	정진만	이름	고성민

[목차]

1. 프로그램 코드 분석

- 다른 숫자나 문자도 입력받을 수 있기에 에러 처리부분을 추가했다.

1.1 프로그램 시작 부분

```
#!/bin/bash
name=Ko_SungMin
student_id=12221828
echo '--------------------------'
echo 'User Name: '$name
echo 'Student Numer: '$student_id
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 data' 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 '--------------------------'
read -p 'Enter your choice [ 1-9 ] ' number
while :
do
```

- 1번째 코드는 내가 사용하려는 명령어 해석기가 bash shell임을 미리 알려준다

- 미리 입력된 이름과 학번을 출력한다.

- 메뉴에 1 ~ 9 번에 해당하는 명령에 대한 설명을 출력한다.

- read -p “” 변수명 구문을 통해 사용자로부터 1 ~ 9 번에 해당하는 숫자를 변수 number로 입력받는다.

- while do ... done문을 이용해서 9번을 입력받기 전까지 프로그램이 계속 작동하도록 한다.

1.2 입력에 따른 1 ~ 9 명령 실행

1.2.1 명령 1

```
23          # 1
24          if [ $number == '1' ] ; then
25              echo ''
26              read -p "Please enter the 'movie id'(1~1682):" movie_id
27              if [ $movie_id -le 1682 ] && [ $movie_id -ge 1 ] ; then
28                  echo ''
29                  awk -F \| '{if($1 == movie_id) print $0}' movie_id=$movie_id u.item
30              elif :; then
31                  error=1
32              fi
```

- number가 1일 때
- read -p '' 변수명 구문으로 사용자로부터 1 ~ 1682사이의 숫자를 변수 movie id로 입력받는다
- 1.2.1.1 movie id가 1 ~ 1682 일 때
- awk 문을 이용해서 |를 구분자로 u.item의 각 라인의 행을 구분하고, 첫번째 행이 movie id와 같을 때 그 라인을 출력한다
- 1.2.1.2 movie id가 1 ~ 1682가 아닐 때 (elif :; then 구문(else를 의미))
- error변수를 1로한다 (예외 처리 부분에서 설명 예정)

1.2.2 명령 2

```
33          # 2
34          elif [ $number == '2' ] ; then
35              echo ''
36              read -p "Do you want to get the data of 'action' genre movies from 'u.item'?(y/n):" agreement
37              if [ $agreement == 'y' ] ; then
38                  echo ''
39                  awk -F \| 'BEGIN {cnt=0} {if ($7 == 1 && cnt < 10){cnt++ ; print $1, $2}}' u.item
40              elif [ $agreement == 'n' ] ; then :;
41              else
42                  error=1
43              fi
```

- number 가 2일 때
- read -p '' 변수명 구문을 통해 agreement 변수를 입력받는다
- 1.2.2.1 agreement가 y일 때
- awk 문을 이용해서 |를 구문자로 u.item의 행을 구분한다.
- 그 후 begin에서 cnt변수를 0이라 선언, u.item의 7번째 행이 의미하는 것이 액션장르이므로 action 장르일때 cnt를 증가시키고 cnt가 10미만일 때 첫번째 행과 2번째행 즉, 영화이름과 영화 제목을 출력한다.
- 1.2.2.2 agreement가 n일 때
- :;를 통해 아무것도 안한다.
- 1.2.2.3 agreement가 y도, n도 아닐 때
- error변수를 1로 한다

1.2.3 명령 3

```
44        # 3
45        elif [ $number == '3' ] ; then
46            echo ''
47            read -p "Please enter the 'movie id'(1~1682):" movie_id
48            if [ $movie_id -le 1682 ] && [ $movie_id -ge 1 ]; then
49                echo ''
50                awk -F' ' 'BEGIN{sum=0; per=0} {if ($2 == movie_id){sum += $3; per +=1}}
51                            END{rate=sum/per; printf "average rating of %d: %1.6g\n", movie_id, rate}' movie_id=$movie_id u.data
52            else
53                error=1
54            fi
```

- read -p '' 변수명 구문을 통해 movie id를 입력받는다
- 1.2.3.1 movie id 가 1 ~ 1682일 때
- awk문을 이용해 u.data에서 공백을 구문자로 행을 구분한다.
- begin부분에서 sum과 per변수를 0으로 선언한다.
- 그 후 2번째 행이 movie id와 같을 때 sum에다 rate를 per에다 1을 더한다
- end부분에서 평점을 계산하고 평점을 소수 6번째 자리에서 반올림하여 출력 포멧에 맞춰서 출력한다.
- 1.2.3.2 movie id 가 1 ~ 1682가 아닐 때
- error변수를 1로 한다

1.2.4 명령 4

```
55          # 4
56          elif [ $number == '4' ] ; then
57              echo ''
58              read -p "Do you want to delete the 'IMDb URL' from 'u.item'?(y/n):" agreement
59              if [ $agreement == 'y' ] ; then
60                  echo ''
61                  cat u.item | sed "s/http.*)//g" | sed -n "1,10p"
62              elif [ $agreement == 'n' ] ; then :;
63              else
64                  error=1
65              fi
```

- number가 4일 때
- read -p '' 변수명 구문을 통해 agreement를 입력받는다.
- 1.2.4.1 agreement가 y일 때
- cat u.item을 통해 u.item 접근 이후 | 을이용해서 sed에 값을 전달한다
- sed "s/변경할값/변경값/g" 구문을 이용해 http로 시작해서)로 끝나는 부분을 공백으로 바꾼다.
- sed 1,10p 구문을 통해 u.user의 1 ~ 10번째 라인을 출력한다
- 1.2.4.2 agreement가 n일 때
- :;을 통해 아무것도 안한다
- 1.2.4.3 agreement가 y도 n도 아닐 때
- error변수를 1로 한다.

1.2.5 명령 5

```
66        # 5
67        elif [ $number == '5' ] ; then
68            echo ''
69            read -p "Do you want to get the data about users from 'u.user'?(y/n)" agreement
70            if [ $agreement == 'y' ] ; then
71                echo ''
72                cat u.user | sed -e "s/^/|/g" -e "s/|/_/g" | sed -e 's/M/male/g' | sed -e 's/F/female/g' -e "s/_/user|/" -e "s/_/|is|/" -e "s/_/|years old|/" -e "s/_/|/" -e "s/_.*//" -e "s/|/ /g" | sed -n "1,10p"
73            elif [ $agreement == 'n' ] ; then :;
74            else
75                error=1
76            fi
```

- number가 5일 때
- read -p '' 변수명 구문을 통해 agreement를 입력받는다.
- 1.2.5.1 agreement가 y일 때
- cat구문을 통해 u.user파일에 접근한다.
- | 를 통해 값을 전달하고 sed 구문을 이용한다.
- (sed -e 구문으로 sed에 여러 명령을 할 수 있다.)
- 처음에 sed s/^/|/g 를통해 앞에 문자 |를 추가하고, s/|/_/g를 통해 |를 _로 변환한다.
- 그후 | 를 통해 값을 전달하고 M을 male로 바꾼후 | 를 통해 값을 전달, F를 female로 전달, 그후 s/바꿀값/변경값/ 구문을 이용해 출력 format을 맞춘다.
- 그 후 |를 통해 값을 전달하고 1 ~ 10까지 한줄 씩 출력한다.
- (s///g = 전체 변경, s/// = 한번 변경)
- 1.2.5.2 agreement가 n일 때
- :;을 통해 아무것도 안한다.
- 1.2.5.3 agreement가 y도 n도 아닐 때
- error변수를 1로 한다

1.2.6 명령 6

```
77        # 6
78        elif [ $number == '6' ] ; then
79            echo ''
80            read -p "Do you want to Modify the format of 'release data' in 'u.item'?(y/n):" agreement
81            if [ $agreement == 'y' ] ; then
82                echo ''
83                cat u.item | sed 's/\(..\-\)\(...\-\)\(....\)/\3\2\1/g' | sed 's/\(.......\)\(\-\)\(..\)\(\-\)/\1\3/g' | sed 's/Jan/01/g' | sed 's/Feb/02/g' | sed 's/Mar/03/g' | sed 's/Apr/04/g' | sed 's/May/05/g' | sed 's/Jun/06/g' |
84            elif [ $agreement == 'n' ] ; then :;
85            else
86                error=1
87            fi
```

```
| sed 's/Jun/06/g' | sed 's/Jul/07/g' | sed 's/Aug/08/g' | sed 's/Sep/09/g' | sed 's/Oct/10/g' | sed 's/Nov/11/g' | sed 's/Dec/012/g' | sed -n "1673,1682p"
```

- number가 6일 때
- read -p '' 변수명 구문을 통해 agreement를 입력받는다.
- 1.2.6.1 agreement가 y일 때
- Cat 구문을 통해 u.item에 접근한다.
- 그 후 | 를 통해 값을 넘겨주고 sed s///g 구문을 이용한다.
- 이때, (..-), (...-), (....)을 ₩1, ₩2, ₩3로 입력받는다.
- (. = 문자하나를 의미한다.)
- 그 후 u.item의 ..-...-.... 문자열이 ....-..-로 변경되게 된다.
- 그리고 | 를 통해 값을 넘겨주고 다시 sed s///g 구문을 이용한다.
- 이때, (.......), (-), (..), (-)을 ₩1, ₩2, ₩3, ₩4로 입력받는다
- 그 후 yyyymmmdd로 변경되게 된다. (y = 년정보, m = 달 정보, d = 일 정보)
- 이 후 | sed s/mmm/숫자/g 구문 여러개를 통해 mmm을 숫자로 변경한다.
- 그 후 sed 1673,1682p 구문을 통해 1673 ~ 1682줄의 영화정보를 출력한다.
- 1.2.6.2 agreement가 n일 때
- :;을 통해 아무것도 안한다.
- 1.2.6.3 agreement가 y도 n도 아닐 때
- error 변수를 1로 한다

1.2.7 명령 7

```
88          # 7
89          elif [ $number == '7' ] ; then
90              echo ''
91              read -p "Please enter the 'user id'(1~943):" user_id
92              if [ $user_id -ge 1 ] && [ $user_id -le 943 ] ; then
93                  echo ''
94                  movie=$(cat u.data | awk '$1 == user_id { print $2 }' user_id=$user_id | sort -n)
95                  #length
96                  length=0
97                  for var in $movie
98                  do
99                      length=$((length+1))
100                 done
101                 #movie_number
102                 cnt=0
103                 out_1=''
104                 for var in $movie
105                 do
106                     if [ $cnt -eq 0 ] ; then
107                         out_1+="$var"
108                     else
109                         out_1+="|$var"
110                     fi
111                     cnt=$((cnt+1))
112                 done
113                 echo $out_1
114                 echo ''
115                 # movie_num, movie_title
116                 cnt=0
117                 for var in $movie
118                 do
119                     if [ $cnt -lt 10 ] ; then
120                         awk -F '|' '$1 == var { printf "%s|%s\n", $1, $2 }' var=$var u.item
121                     fi
122                     cnt=$((cnt+1))
123                 done
124             else
125                 error=1
126             fi
```

- number가 7일 때
- read -p '' 변수명 구문을 통해 user id를 입력받는다.
- 1.2.7.1 user id가 1 ~ 943일 때
- 변수=$(명령어)를 통해 변수를 입력받는다.
- Cat을 통해 u.data에 접근하고 | 를 통해 값을 전달한다.
- 그 후 awk 구문을 이용해 user id와 일치할 때 영화 id를 출력한다.

- 그 후 sort -n 구문을 이용해 출력한 각각의 영화 id를 숫자로 취급하고,
- 오름차순으로 정렬한다.
- 이때 movie변수는 위와 같은 값을 가진 변수가 된다.
- 그 후 movie변수에 대해 for문을 돌며 movie 변수의 길이를 알아낸다.
- length=$((length+1)) (괄호2개 = 산술연산)
- cnt변수=0, out_1=''를 선언한다.
- length만큼 for문을 돌며,
- cnt를 증가시키며 cnt=0일땐 movie id하나(=$var)를 더해주고 아닐땐, |$var를 더해준다.
- 이후 out_1을 출력한다
- 그 후 cnt=0을 선언한다 그후 다시 for문을 돌며 1반복당 cnt를 1회증가시키고, cnt가 10보다 작으면, movie id|movie title 형식으로 출력한다.
- 1.2.7.2 user id가 1 ~ 943이 아닐 때
- error 변수를 1로 한다.

1.2.8 명령 8

```
127         # 8
128         elif [ $number == '8' ] ; then
129             echo ''
130             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):" agreement
131             if [ $agreement == 'y' ] ; then
132                 # 20 <= programmer <= 29 's id
133                 pro_people=$(cat u.user | awk -F '|' '$2 >= 20 && $2 <= 29 { print $0 }' | awk -F '|' '$4 ~ /^programmer/ { print $1 }')
134                 # item length
135                 item_length=$(cat u.item | awk 'BEGIN{cnt=0}{cnt++}END{print cnt}')
136                 # array
137                 movie=()
138                 movie_rate=()
139                 for (( i=0; i<=$item_length; i++))
140                 do
141                     movie+=(0)
142                     movie_rate+=(0)
143                 done
144                 #mean
145                 for person in $pro_people
146                 do
147                     movie_num_rate=$(cat u.data | awk -F ' ' '$1 == person { print $2":"$3 }' person=$person)
148                     for cal in $movie_num_rate
149                     do
150                         movie_num=$(echo $cal | cut -d ':' -f1)
151                         movie_eval=$(echo $cal | cut -d ':' -f2)
152                         movie[$movie_num]=$((${movie[$movie_num]}+1))
153                         movie_rate[$movie_num]=$((${movie_rate[$movie_num]}+$movie_eval))
154                     done
155                 done
156                 echo ''
157                 #print
158                 for (( i=0; i<=$item_length; i++))
159                 do
160                    if [ ${movie[$i]} -gt 0 ] ; then
161                         echo $i ${movie_rate[$i]} ${movie[$i]} | awk '{printf "%1d %1.6g\n", $1, $2/$3}'
162                    fi
163                 done
164             elif [ $agreement == 'n' ] ; then :;
165             else
166                 error=1
167             fi
```

- number가 8일 때
- read -p '' 변수명 구문을 통해 agreement를 입력받는다
- 1.2.8.1 agreement가 y일 때
- 변수명=$(명령어)구문을 통해 pro_people 변수를 입력받는다
- Cat을 통해 u.user에 접근하고 | 를 통해 값을 전달한다.
- 그 후 awk구문을 통해 나이가 20 ~ 29인 user의 정보를 출력받고 | 를 통해 값을 전달한다.
- 그 후 awk구문을 통해 |를 구문자로 구별후 4번째 행이 programmer로 시작할 시 유저의 번호를 출력받는다.
- 이 값이 pro_people의 값이 된다.
- 이 후 변수명=$(명령어) 구문을 통해 item_length를 얻는다.

- 먼저 cat으로 u.item으로 접근하고 | 로 값을 전달한다.
- Awk 구문을 이용해 begin cnt=0을 선언후 cnt를 증가시키며 end에서 cnt를 출력받는다.
- 이 값이 item_length의 값이 된다.
- 이후 movie, movie_rate를 item_length만큼 for문을 돌며, 어레이를 선언한다.
- 이후 pro_people의 값에 대해 for문을 돌며, movie_id:movie_rate 꼴로 값을 받는다.
- 이후 그 값에 대해 for문을 돌며, 변수명=$(명령어)구문을 통해
- movie_num= cut -d를 이용해 ':'를 구문자로 자른것의 앞부분(movie_id)을, movie_eval= cut -d를 이용해 ':'를 구문자로 자른것의 뒷부분(movie_rate)을 변수로 받는다.
- 그 후 movie[movie_id] = 기존값 + 1, movie_rate[movie_id] = 기존값 + movie_rate를 해준다.
- 2중 for문을 마친 후 다시 item_length만큼 for문을 돌며(이때, i변수 사용), 평가한 사람이 아무도 없는 영화를 제외하고 print format에 맞춰 movie_rate[i](총 평가 rate)/movie[i](총 평가 사람수)를 소수점 6번째 자리에서 반올림하여 "movie id movie rate"꼴로 출력한다.
- **이때 2중 for문 및 for문을 여러 번 돌기 때문에 실행시간이 오래 걸린다. 약 1분정도**
- 1.2.8.2 agreement가 n일 때
- ;;을 통해 아무것도 안한다
- 1.2.8.3 agreement가 y도 n도 아닐 때
- error변수를 1로 한다

1.2.9 명령 9

```
168         # 9
169         elif [ $number == '9' ] ; then
170             echo 'Bye!'
171             break
```

- Number가 9일 때
- Bye!를 출력하고
- Break를 이용해 while문을 탈출하고 프로그램이 종료된다

1.3 에러 처리

1.3.1 에러 변수 선언

```
21          # except value
22          error=0
```

error변수는 초기값이 0이다

1.2에따라 각 명령을 실행시키며 error의 값이 1로 변경되기도 한다.

1.3.2 number가 범위를 벗어난 숫자 or 문자일때, error 상황일 때

```
172         # 10 except situation
173         else
174             error=1
175         fi
176         # if error
177         if [ $error -eq 1 ] ; then
178             echo ''
179             echo error occurred!
180             echo ''
181             read -p 'Enter your choice [ 1-9 ] ' number
182         else
183             echo ''
184             read -p 'Enter your choice [ 1-9 ] ' number
185         fi
186     done
```

- error=1 이라는 의미는 프로그램 진행중 범위를 벗어난 문자, 범위를 벗어난 숫자를 입력받았을 때이다.
- 이때, error occurred!를 출력하고, number를 다시 입력받는다.
- error=0 이라는 의미는 프로그램이 정상적으로 진행됐음을 의미한다.
- 이때는 명령어에 해당하는 명령을 수행 후 다시 number를 입력받는다.(number=9 제외)

2. 프로그램 실행결과

2.1 프로그램 시작 부분

```
rokorori@rokorori-VirtualBox:~$ ./prj1_12221828_kosungmin.sh
--------------------------
User Name: Ko_SungMin
Student Numer: 12221828
[ MENU ]
1. Get the data of the movie identified by a specific 'movie id' from 'u.item'
2. Get the data of action genre movies from 'u.item'
3. Get the average 'rating' of the movie identified by specific 'movie id' from 'u.data'
4. Delete the 'IMDb URL' from 'u.item'
5. Get the data about users from 'u.user'
6. Modify the format of 'release data' in 'u.item'
7. Get the data of movies rated by a specific 'user id' from 'u.data'
8. Get the average 'rating' of movies rated by users with 'age' between 20 and 29 and 'o
ccupation' as 'programmer'
9. Exit
--------------------------
Enter your choice [ 1-9 ]
```

- 이름과 학번을 출력한다
- 메뉴에 1 ~ 9 번에 해당하는 명령을 설명하고 출력한다
- 사용자로부터 1 ~ 9의 명령을 입력받는다

2.2 명령 1

```
Enter your choice [ 1-9 ] 1

Please enter the 'movie id'(1~1682):1

1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Toy%20Story%20(
1995)|0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0

Enter your choice [ 1-9 ]
```

- 사용자로부터 1 ~ 1682 사이의 movie id를 입력 받는다.
- 해당하는 movie의 정보를 출력한다.

2.3 명령 2

2.3.1 사용자로부터 y를 입력 받았을 때

```
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 ]
```

- u.item으로부터 action 장르 영화 10개를 출력한다.
- 사용자로부터 1 ~ 9의 숫자를 입력받는다.

2.3.2 사용자로부터 n을 입력 받았을 때

```
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 ]
```

- 사용자로부터 1 ~ 9의 숫자를 입력받는다.

2.4 명령 3

```
Enter your choice [ 1-9 ] 3

Please enter the 'movie id'(1~1682):1

average rating of 1: 3.87832

Enter your choice [ 1-9 ]
```

- 사용자로부터 1 ~ 1682 사이의 movie id를 입력 받는다.
- Movie의 평균 rate를 출력한다
- 사용자로부터 1 ~ 9의 숫자를 입력받는다.

2.5 명령 4

2.5.1 사용자로부터 y를 입력 받았을 때

```
Enter your choice [ 1-9 ] 4

Do you want to delete the 'IMDb URL' from 'u.item'?(y/n):y

1|Toy Story (1995)|01-Jan-1995|||0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0
2|GoldenEye (1995)|01-Jan-1995|||0|1|1|0|0|0|0|0|0|0|0|0|0|0|1|0|0
3|Four Rooms (1995)|01-Jan-1995|||0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
4|Get Shorty (1995)|01-Jan-1995|||0|1|0|0|0|1|0|0|1|0|0|0|0|0|0|0|0|0
5|Copycat (1995)|01-Jan-1995|||0|0|0|0|0|0|1|0|1|0|0|0|0|0|0|1|0|0
6|Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)|01-Jan-1995|||0|0|0|0|0|0
|0|0|1|0|0|0|0|0|0|0|0|0
7|Twelve Monkeys (1995)|01-Jan-1995|||0|0|0|0|0|0|0|0|1|0|0|0|0|0|1|0|0|0
8|Babe (1995)|01-Jan-1995|||0|0|0|0|1|1|0|0|1|0|0|0|0|0|0|0|0|0
9|Dead Man Walking (1995)|01-Jan-1995|||0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|0|0
10|Richard III (1995)|22-Jan-1996|||0|0|0|0|0|0|0|0|1|0|0|0|0|0|0|0|1|0

Enter your choice [ 1-9 ]
```

- u.item으로부터 IMDb URL을 삭제한 영화 1 ~ 10번의 정보를 출력한다.
- 사용자로부터 1 ~ 9의 숫자를 입력받는다.

2.5.2 사용자로부터 n을 입력 받았을 때

```
Enter your choice [ 1-9 ] 4

Do you want to delete the 'IMDb URL' from 'u.item'?(y/n):n

Enter your choice [ 1-9 ]
```

- 사용자로부터 1 ~ 9의 숫자를 입력받는다.

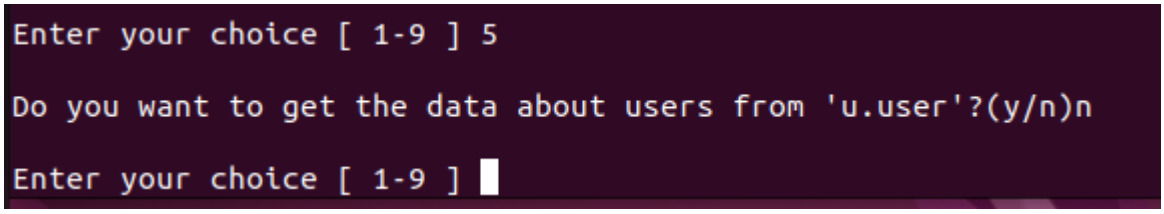
2.6 명령 5

2.6.1 사용자로부터 y를 입력 받았을 때

```
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 ]
```

- u.user로부터 1 ~ 10번의 유저정보를 위 이미지의 포멧으로 출력한다.
- 사용자로부터 1 ~ 9번의 숫자를 입력받는다.

2.6.2 사용자로부터 n을 입력 받았을 때



- 사용자로부터 1 ~ 9번의 숫자를 입력받는다.

2.7 명령 6

2.7.1 사용자로부터 y를 입력 받았을 때

- u.item으로부터 1673 ~ 1682번의 영화의 release data를 YYMMDD의 형식으로 출력한다.

- 사용자로부터 1 ~ 9번의 숫자를 입력받는다.

2.7.2 사용자로부터 n을 입력 받았을 때

```
Enter your choice [ 1-9 ] 6

Do you want to Modify the format of 'release data' in 'u.item'?(y/n):n

Enter your choice [ 1-9 ]
```

- 사용자로부터 1 ~ 9번의 숫자를 입력받는다.

2.8 명령 7

```
Enter your choice [ 1-9 ] 7

Please enter the 'user id'(1~943):1

1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|16|17|18|19|20|21|22|23|24|25|26|27|28|29|30|31|32|33|34|35|
36|37|38|39|40|41|42|43|44|45|46|47|48|49|50|51|52|53|54|55|56|57|58|59|60|61|62|63|64|65|66|67|
68|69|70|71|72|73|74|75|76|77|78|79|80|81|82|83|84|85|86|87|88|89|90|91|92|93|94|95|96|97|98|99|
100|101|102|103|104|105|106|107|108|109|110|111|112|113|114|115|116|117|118|119|120|121|122|123|
124|125|126|127|128|129|130|131|132|133|134|135|136|137|138|139|140|141|142|143|144|145|146|147|
148|149|150|151|152|153|154|155|156|157|158|159|160|161|162|163|164|165|166|167|168|169|170|171|
172|173|174|175|176|177|178|179|180|181|182|183|184|185|186|187|188|189|190|191|192|193|194|195|
196|197|198|199|200|201|202|203|204|205|206|207|208|209|210|211|212|213|214|215|216|217|218|219|
220|221|222|223|224|225|226|227|228|229|230|231|232|233|234|235|236|237|238|239|240|241|242|243|
244|245|246|247|248|249|250|251|252|253|254|255|256|257|258|259|260|261|262|263|264|265|266|267|
268|269|270|271|272

1|Toy Story (1995)
2|GoldenEye (1995)
3|Four Rooms (1995)
4|Get Shorty (1995)
5|Copycat (1995)
6|Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)
7|Twelve Monkeys (1995)
8|Babe (1995)
9|Dead Man Walking (1995)
10|Richard III (1995)

Enter your choice [ 1-9 ]
```

- 유저가 평가한 영화의 정보를 위와 같은 format으로 출력한다.
- 유저가 평가한 영화중 movie id가 작은 10개를 movie id|movie title의 형식으로 출력한다.
- 사용자로부터 1 ~ 9번의 숫자를 입력받는다.

2.9 명령 8

2.9.1 사용자로부터 y를 입력 받았을 때

```
Enter your choice [ 1-9 ] 8

Do you want to get the average 'rating' of movies rated by users with 'age' between 20 and 29 an
d '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
```

```
1491 1
1509 1
1512 3
1513 2
1518 4
1531 3
1552 2
1597 1
1600 4
1621 1
1655 2

Enter your choice [ 1-9 ]
```

- 나이가 20 ~ 29이고, 직업이 프로그래머인 유저들만으로 평가 받은 영화 평점을 모두 출력한다.
- 사용자로부터 1 ~ 9번의 숫자를 입력받는다.

2.9.2 사용자로부터 n을 입력 받았을 때

```
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):n

Enter your choice [ 1-9 ]
```

- 사용자로부터 1 ~ 9번의 숫자를 입력받는다.

2.10 명령 9

```
Enter your choice [ 1-9 ] 9
Bye!
rokorori@rokorori-VirtualBox:~$
```

- Bye!를 출력한다.
- 프로그램이 종료된다.

2.11 에러 처리

2.11.1 사용자로부터 잘못된 입력을 받았을 때

```
error occurred!

Enter your choice [ 1-9 ]
```

- "error occurred!" 를 출력한다
- 사용자로부터 1 ~ 9번의 숫자를 입력받는다

2.11.1.1 초기 메뉴에서 잘못된 입력을 받았을 때

```
Enter your choice [ 1-9 ] 0

error occurred!

Enter your choice [ 1-9 ] 10

error occurred!

Enter your choice [ 1-9 ] k

error occurred!

Enter your choice [ 1-9 ]
```

2.11.1.2 명령 1에서 잘못된 입력을 받았을 때

```
Enter your choice [ 1-9 ] 1

Please enter the 'movie id'(1~1682):0

error occurred!

Enter your choice [ 1-9 ] 1

Please enter the 'movie id'(1~1682):1683

error occurred!

Enter your choice [ 1-9 ] 1

Please enter the 'movie id'(1~1682):k
./prj1_12221828_kosungmin.sh: line 27: [: k: integer expression expected

error occurred!

Enter your choice [ 1-9 ]
```

2.11.1.3 명령 2에서 잘못된 입력을 받았을 때

```
Enter your choice [ 1-9 ] 2

Do you want to get the data of 'action' genre movies from 'u.item'?(y/n):2

error occurred!

Enter your choice [ 1-9 ] 2

Do you want to get the data of 'action' genre movies from 'u.item'?(y/n):k

error occurred!

Enter your choice [ 1-9 ]
```

2.11.1.4 명령 3에서 잘못된 입력을 받았을 때

```
Enter your choice [ 1-9 ] 3

Please enter the 'movie id'(1~1682):0

error occurred!

Enter your choice [ 1-9 ] 3

Please enter the 'movie id'(1~1682):1683

error occurred!

Enter your choice [ 1-9 ] 3

Please enter the 'movie id'(1~1682):k
./prj1_12221828_kosungmin.sh: line 48: [: k: integer expression expected

error occurred!

Enter your choice [ 1-9 ]
```

2.11.1.5 명령 4에서 잘못된 입력을 받았을 때

```
Enter your choice [ 1-9 ] 4

Do you want to delete the 'IMDb URL' from 'u.item'?(y/n):1

error occurred!

Enter your choice [ 1-9 ] 4

Do you want to delete the 'IMDb URL' from 'u.item'?(y/n):k

error occurred!

Enter your choice [ 1-9 ]
```

2.11.1.6 명령 5에서 잘못된 입력을 받았을 때

```
Enter your choice [ 1-9 ] 5

Do you want to get the data about users from 'u.user'?(y/n)1

error occurred!

Enter your choice [ 1-9 ] 5

Do you want to get the data about users from 'u.user'?(y/n)k

error occurred!

Enter your choice [ 1-9 ]
```

2.11.1.7 명령 6에서 잘못된 입력을 받았을 때

```
Enter your choice [ 1-9 ] 6

Do you want to Modify the format of 'release data' in 'u.item'?(y/n):1

error occurred!

Enter your choice [ 1-9 ] 6

Do you want to Modify the format of 'release data' in 'u.item'?(y/n):k

error occurred!

Enter your choice [ 1-9 ]
```

2.11.1.8 명령 7에서 잘못된 입력을 받았을 때

```
Enter your choice [ 1-9 ] 7

Please enter the 'user id'(1~943):0

error occurred!

Enter your choice [ 1-9 ] 7

Please enter the 'user id'(1~943):944

error occurred!

Enter your choice [ 1-9 ] k

error occurred!

Enter your choice [ 1-9 ]
```

2.11.1.9 명령 8에서 잘못된 입력을 받았을 때

```
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):k

error occurred!

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):1

error occurred!

Enter your choice [ 1-9 ]
```

3. 참조

https://m.blog.naver.com/eleexpert/140133310421 (#!/bin/bash의 의미)

https://stackoverflow.com/questions/3224878/what-is-the-purpose-of-the-colon-gnu-bash-builtin (if :;의 의미)

https://lhc9763.tistory.com/9 (bash shell script if문)