

```

1  #! /bin/sh
2
3  #declare and fill the four suits for the deck
4  declare -a spades=('Ace_of_Spades' 'King_of_Spades' \
5                    'Queen_of_Spades' 'Jack_of_Spades' \
6                    '10_of_Spades' '9_of_Spades' \
7                    '8_of_Spades' '7_of_Spades' \
8                    '6_of_Spades' '5_of_Spades' \
9                    '4_of_Spades' '3_of_Spades' \
10                   '2_of_Spades')
11
12  declare -a diamonds=('Ace_of_Diamonds' 'King_of_Diamonds' \
13                     'Queen_of_Diamonds' 'Jack_of_Diamonds' \
14                     '10_of_Diamonds' '9_of_Diamonds' \
15                     '8_of_Diamonds' '7_of_Diamonds' \
16                     '6_of_Diamonds' '5_of_Diamonds' \
17                     '4_of_Diamonds' '3_of_Diamonds' \
18                     '2_of_Diamonds')
19
20  declare -a clubs=('Ace_of_Clubs' 'King_of_Clubs' \
21                  'Queen_of_Clubs' 'Jack_of_Clubs' \
22                  '10_of_Clubs' '9_of_Clubs' \
23                  '8_of_Clubs' '7_of_Clubs' \
24                  '6_of_Clubs' '5_of_Clubs' \
25                  '4_of_Clubs' '3_of_Clubs' \
26                  '2_of_Clubs')
27
28  declare -a hearts=('Ace_of_Hearts' 'King_of_Hearts' \
29                   'Queen_of_Hearts' 'Jack_of_Hearts' \
30                   '10_of_Hearts' '9_of_Hearts' \
31                   '8_of_Hearts' '7_of_Hearts' \
32                   '6_of_Hearts' '5_of_Hearts' \
33                   '4_of_Hearts' '3_of_Hearts' \
34                   '2_of_Hearts')
35
36  #declare the play suits
37  declare -a play_spades=()
38  declare -a play_diamonds=()
39  declare -a play_clubs=()
40  declare -a play_hearts=()
41
42  #function to shuffle the deck
43  new_deck () {
44      spades_gone=0;diamonds_gone=0;clubs_gone=0;hearts_gone=0
45      play_spades=(${spades[*]})
46      play_diamonds=(${diamonds[*]})
47      play_clubs=(${clubs[*]})
48      play_hearts=(${hearts[*]})
49  }
50
51  #function to draw cards

```

```

52 get_card () {
53     clear
54     #check for valid input
55     read -p "How many cards would you like to draw from this deck? "
... cards_requested
56     if ! [[ $cards_requested =~ ^[+]?[0-9] ]] 2>>/dev/null;then
57         clear
58         echo "Invalid option. Press Enter to return to the main menu: "
59         read
60         return
61     fi
62
63     #see if there are enough cards in the deck to meet the request
64     cards_remaining=$(( ${#play_spades[@]} + ${#play_diamonds[@]} +
... ${#play_clubs[@]} + ${#play_hearts[@]})
65     if [[ $cards_requested -gt $cards_remaining ]];then
66         clear
67         echo -e "There are only $cards_remaining cards left in the deck but
... you have requested $cards_requested cards.\n"
68         echo "Press the Enter key to return to the main menu: "
69         read
70         return
71     fi
72     echo -e "Your cards are:\n"
73
74     #loop and get the number of requested cards
75     while [[ $cards_requested -gt 0 ]];
76     do
77         #check to see if all suits are empty
78         if [[ spades_gone -eq 1 && diamonds_gone -eq 1 && clubs_gone -eq 1
... && hearts_gone -eq 1 ]];then
79             echo "all done"
80             break
81         fi
82
83         #get a random suit
84         suit=$((RANDOM%4))
85         if [[ $suit -eq 0 ]];then #get a spade
86             suit_length=${#play_spades[@]}
87             if [[ $suit_length -eq 0 ]]; then
88                 spades_gone=1
89                 continue
90             else
91                 card=$((RANDOM%suit_length))
92                 echo ${play_spades[$card]}
93                 play_spades=( ${play_spades[*]:0:$card}
... ${play_spades[*]:$(( $card + 1 ))})
94                 cards_requested=$(( $cards_requested-1))
95             fi
96
97         elif [[ $suit -eq 1 ]];then #get a diamond

```

```

98         suit_length=${#play_diamonds[@]}
99         if [[ $suit_length -eq 0 ]]; then
100             diamonds_gone=1
101             continue
102         else
103             card=$((RANDOM%suit_length))
104             echo ${play_diamonds[$card]}
105             play_diamonds=(${play_diamonds[*]:0:$card}
... ${play_diamonds[*]:$(( $card + 1 ))})
106             cards_requested=$(( $cards_requested - 1 ))
107         fi
108     elif [[ $suit -eq 2 ]]; then #get a club
109         suit_length=${#play_clubs[@]}
110         if [[ $suit_length -eq 0 ]]; then
111             clubs_gone=1
112             continue
113         else
114             card=$((RANDOM%suit_length))
115             echo ${play_clubs[$card]}
116             play_clubs=(${play_clubs[*]:0:$card}
... ${play_clubs[*]:$(( $card + 1 ))})
117             cards_requested=$(( $cards_requested - 1 ))
118         fi
119     else
120         suit_length=${#play_hearts[@]}
121         if [[ $suit_length -eq 0 ]]; then #get a heart
122             hearts_gone=1
123             continue
124         else
125             card=$((RANDOM%suit_length))
126             echo ${play_hearts[$card]}
127             play_hearts=(${play_hearts[*]:0:$card}
... ${play_hearts[*]:$(( $card + 1 ))})
128             cards_requested=$(( $cards_requested - 1 ))
129         fi
130     fi
131 done
132 echo
133 read -p "Press Enter to continue"
134 }
135
136 #get the new deck for the first time
137 new_deck
138
139 #menu
140 while true;
141 do
142
143     clear
144     echo "
145     Welcome to the card deck simulator.

```

```
146
147 Please select from the following options:
148
149     1. Draw a selected number of cards from the current deck
150     2. Get a new deck of cards
151     3. Exit
152 "
153     read -p "Option#: " user_menu_choice
154
155     if [[ $user_menu_choice -eq 1 ]];then
156         get_card
157     elif [[ $user_menu_choice -eq 2 ]];then
158         new_deck
159     elif [[ $user_menu_choice -eq 3 ]];then
160         clear
161         break
162     else
163         read -p "That is not a valid selection. Press Enter to continue: "
164     fi
165 done
166
167
168
169
170
171
172
173
174
175
176
177
```