Emre ÖZDİL
150120138

Istanbul Technical University
Department of Computer Engineering

BLG 233E - Data Structures
Assignment 2
Report

My main function has one declaration and seven significant function.

Firstly, I create deck array

```
int main()
{
    int deck[53];

    create();
    deck_create(deck);
    deck_shuffle(deck,counter);
    add_queue(deck);
    operation();
    result();
    close();
}
```

I assign sorted deck to array.

```
void deck_create(int *d)
{
    for (int i = 1; i < 53; i++)
        d[i] = i;
}
```

Pseudo Code of Recursive Shuffling Algorithm

Function deck_shuffle (Deck array, counter)
    Initialize a random number (1-52);
    Swap(Deck[counter], Deck[random]);
    if counter=52 **then** end function;
    deck_shuffle (Deck array, counter+1);

Swap function
    a=a+b;
    b=a-b;
    a=a-b;

```
void deck_shuffle(int *d,int c)
{
    int random = (rand() % 52) + 1;
    /*
    Change variable
    Like
    (a,b)->(b,a):
        a=a+b
        b=a-b
        a=a-b
    (b,a)
    */
    d[c] = d[c] + d[random];
    d[random] = d[c] - d[random];
    d[c] = d[c] - d[random];
    if (c == 52)    return;//end of recursion
    deck_shuffle(d, c + 1);
}
```

Shuffling deck add queue.

```
void add_queue(int *d)
{
    fprintf(out, "Shuffled deck:\n");
    for (int i = 1; i < 53; i++)//add queue
        que.enqueue(d[i]);
    for (int i = 1; i < 53; i++)//print deck
    {
        fprintf(out,"%d:", i);
        print(d[i]);
        fprintf(out,"\n");
    }
}
```

Operation function include while and for loop check cards.

The temporary space which is used to stack the cards that do not match during the counting & checking operation using the Stack data structure.
Comparing the card from the beginning of the queue with the counting number and placing the card to the appropriate place.
To fortune-telling cards

```
if (i == temp % 13 || (i == 13 && temp % 13 == 0))
{
    //match print
    fortune.enqueue(temp);//add fortune queue
    st.pop();//match card
    while (!st.isempty())
        que.enqueue(st.pop());//add deck stack->queue
    fprintf(out,"%d:", i);
    print(temp);
    fprintf(out,":match\n");
    break;
}
```

Determining the end of counting when no more cards match with the counting number

```
if (que.isempty())  break;//Determining the end of counting
```

Summing loop

```
while (!fortune.isempty())
{
    temp = fortune.dequeue();
    if (temp % 13 == 0)
        sum += 13;
    else
        sum += temp % 13;
    print(temp);
    fprintf(out,"\n");
}
```

My most important function in order to write output is print function.
It has two switch case statement so as to determine card type.

SSH Compiling

```
[ozdile@ssh blg233e]$ cd hw3
[ozdile@ssh hw3]$ ls
main.cpp  queue.cpp  queue.h  README.txt  stack.cpp  stack.h
[ozdile@ssh hw3]$ g++ *.cpp
[ozdile@ssh hw3]$ ls
a.out  main.cpp  queue.cpp  queue.h  README.txt  stack.cpp  stack.h
[ozdile@ssh hw3]$ ./a.out
[ozdile@ssh hw3]$ ls
a.out     output.txt  queue.h     stack.cpp
main.cpp  queue.cpp   README.txt  stack.h
[ozdile@ssh hw3]$
```