

## LAB ASSIGNMENT 10

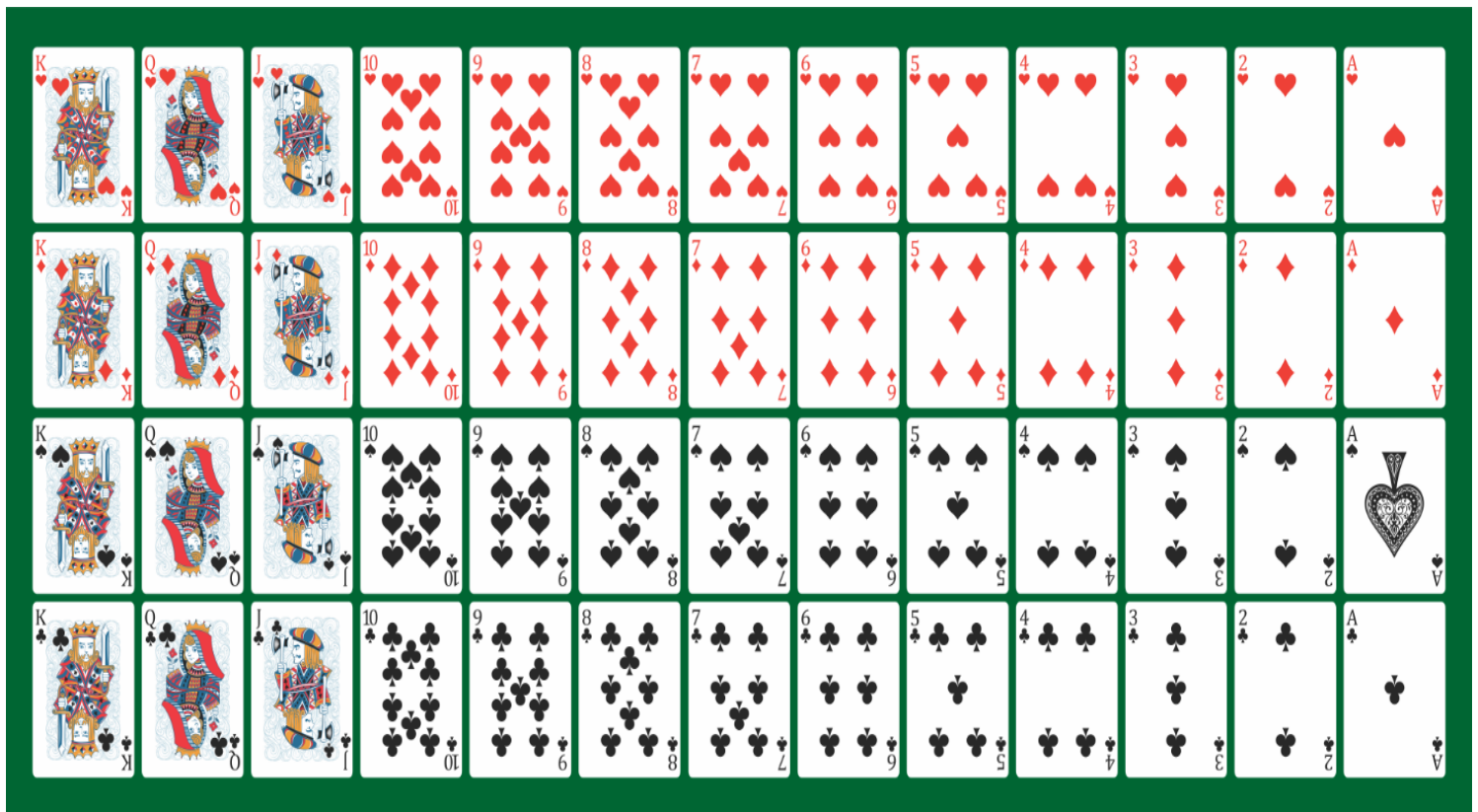
- Write a program **maxmin.c** to find the largest and smallest elements in an array using a fixed function call (`max_min()`) that involves both call by addressing and call by value.
  - Modify the **maxmin.c** program so that the `max_min()` function uses a pointer instead of an integer to keep track of the current position in the array.
- It says that there is always one prime number between any two consecutive natural numbers' square ( $N = 1, 2, 3, \dots$ ). This is called **Legendre's Conjecture**. Conjecture: A conjecture is a proposition or conclusion based upon incomplete information to which no proof has been found i.e it has not been proved or disproved. Write a C program to verify Legendre's Conjecture for a given N.
- Write a C program `deal.c` that deals a random hand from a standard deck of playing cards, i.e., we pick cards randomly from the deck and avoid picking the same card twice. (In case you haven't had time to play games recently, each card in a standard deck has a **suit** – clubs (c), diamonds (d), hearts (h), or spades (s) – and a **rank** – two (2), three (3), four (4), five (5), six (6), seven (7), eight (8), nine (9), ten (10), jack (J), queen (Q), king (K), or ace (A).) We'll have the user to specify how many cards should be in hand, for example:

Enter number of cards in hand: 5

And the output should be as follows:

Your hand: 7c 2s 5d As 2h

(**Hint:** use `srand()` and `rand()` functions from `<stdlib.h>` and `time()` function from `<time.h>`)



4. Modify the deal.c program of Q.No. 6 so that it prints the full names of the cards it deals:

```
Enter number of cards in hand: 5
You hand:
Seven of clubs
Two of spades
Five of diamonds
Ace of spades
Two of hearts
```

5. A Credit Card number is usually a 16-digit number. A valid Credit Card number would satisfy the Luhn's algorithm explained below with the help of a dummy Credit Card number – 4567 1234 5678 9129.

- Start with the second to last digit and multiply every other digit by 2.

4	5	6	7	1	2	3	4	5	6	7	8	9	1	2	9
8		12		2		6		10		14		18		4	

- Then subtract 9 from any number larger than 10. Thus we get:

8	3	2	6	1	5	9	4
---	---	---	---	---	---	---	---

- Add them all up to get 38.
- Add all the other digits to get 42.
- Sum of 38 and 42 is 80. Since 80 is divisible by 10, the Credit Card number is valid.

Write a C program that receives a Credit Card number and checks using the above rule whether the Credit Card number is valid.

6. (a) Write a C program that reads a message, then prints the reversal of the message:

```
Enter a message: Don't get mad, get even.
Reversal is: .neve teg ,dam teg t'noD
```

**Hint:** Read the message one character at a time (using getchar) and store the characters in an array. Stop reading when the array is full or the character read is '\n'.

(b) Revise the program to use a pointer instead of an integer to keep track of the current position in the array.

7. (a) Write a C program that reads a message, then checks whether it's a palindrome (the letters in the message are the same from left to right as from right to left):

```
Enter a message: He lived as a devil, eh?
Palindrome
```

```
Enter a message: Madam, I am Adam.
Not a palindrome
```

Ignore all characters that aren't letters. Use integer variables to keep track of positions in the array.

**(b)** Revise the program to use pointers instead of integers to keep track of positions in the array.

8. Write a program that will read a line and delete from it all occurrences of the word '**the**'.