Programming For Engineers Laboratory

Course ID: EE058IU

Lab 5

Working with Pointer

Full name + Student ID: ……

……

Class: ……

Group: ……

Date: ……

I. Objectives

Understanding the use of pointer in C programming. Why do we need to use pointer and how to use pointer to solve specific problems.

Get familiar with the skill of working with pointer.

II. Pre-Lab Preparation

Read the theory carefully at home before coming to the class.

III. In-Lab Procedure

Exercise 1

Write a C program to process arithmetic operations (Sum, difference, product, quotient) using **pointers**.

Output:

Enter two real numbers: 20 5
Sum = 25.00
Difference = 15.00
Product = 100.00
Quotient = 4.00

Exercise 2

Imagine that you are writing a mini game that tracks the location of a ship using latitudes and longitudes. If a player wants to travel southeast, that means her latitude will go down and her longitude will go up. So now you could write a `go_south_east()` function that takes arguments for latitude and longitude, which will then decrease and increase.

a. Run the following program. Show your final latitude and longitude after calling the *go_south_east()* function.? Do you expect that result? Explain why/why not.

```c
#include <stdio.h>
void go_south_east(int lat, int lon){
    lat = lat - 1;
    lon = lon + 1;
}
int main(){
    int latitude = 32;
    int longitude = -12;
    go_south_east(latitude, longitude);
    printf("Let's go South East! We now at [%i, %i]\n",latitude, longitude);
return 0;
}
```

b. Now you need to fix the *go_south_east()* function so that it uses pointers to update the correct data. Show your final program and comments on what you fixed?

Exercise 3

Write a C program that reads through an array using pointer. Then, scan through this array to find a particular value **(using the pointer).**

Output:

Enter a size of array: 6

Enter element a[0]: 5

Enter element a[1]: 3

Enter element a[2]: 7

Enter element a[3]: 9

Enter element a[4]: 5

Enter element a[5]: 6

5 3 7 9 5 6

Enter a search key: 8

No position was found !!!

Enter a search key: 6

The search key is in the position : 6

Exercise 4a.

Write a C program that generates an array (50 elements) with random numbers from 1 to 50. Then the user will input an integer, and the program should count how many times that number occurs in the array and delete all elements equal to the input number from array. **Students must use pointers**.

Output:

The original array

```
10   2  39  23  18  22  29  16  29  28
35   1   3  15  22   3  14  24  13  48
16  10  45  18  49  12  22  36   7  18
 1  15  44  23   2  47  36  28   4  18
 2  20  33  19  11  24  27  32  13  27
```

Enter a value (from 1 to 50) to be deleted = 22

The new array

```
10   2  39  23  18  29  16  29  28  35
 1   3  15   3  14  24  13  48  16  10
45  18  49  12  36   7  18   1  15  44
23   2  47  36  28   4  18   2  20  33
19  11  24  27  32  13  27
```

There are 3 values of 22 being deleted from the original array/

Exercise 4b

Continue with exercise 4a. Using function pointer to allow user to choose whether to sort the array in ascending order or descending order. Print the sorted array. **Students must use pointer(s) in sorting functions**.

Question 5: Return multiple values from function using pointers.

Write a program to return Min and Max value of an array. The array size and array's elements are inputted by user. Your program should have a function to get the Min and Max element using pointer.

Question 6: What is the purpose of the following C code. Assume that you are a senior programmer who will deliver this code to a fresher. Make sufficient comments so that the fresher understands your code. Use // for single-line comment and /* */ for block comment.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define SUITS 4
#define FACES 13
#define CARDS 52

void shuffle(int deck[][FACES]);
void deal(int deck[][FACES], const char *face[], const char *suit[]);

int main(void) {
    int deck[SUITS][FACES] = {0};
    srand(time(NULL));
    shuffle(deck);
    const char *suit[SUITS] = {"Hearts", "Diamonds", "Clubs", "Spades"};
    const char *face[FACES] = {"Ace", "Deuce", "Three", "Four", "Five",
        "Six", "Seven", "Eight", "Nine", "Ten", "Jack", "Queen", "King"};
    deal(deck, face, suit);
}
void shuffle(int deck[][FACES]) {
    for (size_t card = 1; card <= CARDS; ++card) {
        size_t row = 0;
        size_t column = 0;

        do {
            row = rand() % SUITS;
            column = rand() % FACES;
        } while(deck[row][column] != 0);
        deck[row][column] = card;
    }
}
void deal(int deck[][FACES], const char *face[], const char *suit[]) {
    for (size_t card = 1; card <= CARDS; ++card) {
        for (size_t row = 0; row < SUITS; ++row) {
            for (size_t column = 0; column < FACES; ++column) {
                if (deck[row][column] == card) {
                    printf("%5s of %-8s %c", face[column], suit[row],
                        card % 4 == 0 ? '\n' : '\t');
                }
            }
        }
    }
}
```

Question 7: Modify the program in Question 6 so that the card-dealing function deals a five-card poker hand. Then write the following in addition functions:

a. Determine whether the hand contains a pair.
b. Determine whether the hand contains two pairs.
c. Determine whether the hand contains three of a kind (e.g., three jacks).
d. Determine whether the hand contains four of a kind (e.g., four aces).
e. Determine whether the hand contains a flush (i.e., all five cards of the same suit).
f. Determine whether the hand contains a straight (i.e., five cards of consecutive face values).

Your main program should be able to display the result.
Reference: https://en.wikipedia.org/wiki/List_of_poker_hands

THE END