This is individual coursework. Time: 2 Hours

You cannot use Pointers in this CW.

Expectations: each expression/statement has proper comment(s) to explain it.
Use the minimum efforts to achieve the desired results.

Submit in one text file
StudentID_code.txt (replace student id with your own student id number, put your codes in that .txt file, should be separated with question no).

Late submission: In case you face problem in uploading at Moodle, email me code file (.txt) no later than 1 minute delay (10:01 PM), and then upload it on Moodle within 15 minutes (10:15 PM).
Failing to email me the code till 10:01 PM, first 5 minutes delay (10:05 PM), -1 mark and then every 1-minute delay, 1 more mark will be deducted.

There will be no compromise on any kind of cheating or plagiarism. This is an individual coursework.

Q1. The following program generates 100 random numbers and arrange them in ascending or descending order based on input A (for ascending) or D (for descending).

You have to call the functions from the `main()` function, mentioned in the prototype, and display the results when input A (or a) for ascending and D (or d) for descending. You cannot change the original array, also correct the syntax if necessary. Use the minimum efforts to achieve the desired results. (The following code is also available with just after this file on Moodle).

```
void ascending(const int a[]);
void descending(const int a[]);

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define SIZE 100

int main(void) {

     int a[SIZE] = {0};
     srand(time(NULL));
    for (size_t i = 0; i<SIZE; ++i){
     a[i] = 1+rand() %SIZE;
    }

    puts("Data items in original order");

    for (size_t i = 0; i < SIZE; ++i) {
       printf("%4d", a[i]);
```

```
    }

        char g;
        printf("\n Enter A for Ascending and D for
Descending: ");
        scanf("%c",&g);

    switch(g)
    {
    case 'A':
    case ‘a’:
        {
// Ascending
    for (int pass = 1; pass < SIZE; ++pass) {
       for (size_t i = 0; i < SIZE - 1; ++i) {
          if (a[i] > a[i + 1]) {
             int hold = a[i];
             a[i] = a[i + 1];
             a[i + 1] = hold;
          }
       }
    }
    puts("\nData items in ascending order");
    for (size_t i = 0; i < SIZE; ++i) {
       printf("%4d", a[i]);
    }
    puts("");
    break;
    }
    case 'D':
    case ‘d’:
    {
       // Descending
       for (int pass = 1; pass < SIZE; ++pass) {
          for (size_t i = 0; i < SIZE - 1; ++i) {
             if (a[i] < a[i + 1]) {
                int hold = a[i];
                a[i] = a[i + 1];
                a[i + 1] = hold;
             }
          }
       }
       puts("\nData items in descending order");
       for (size_t i = 0; i < SIZE; ++i) {
          printf("%4d", a[i]);
       }
       puts("");
       break;}
       }
       return 0;
    }
```

[marks: 10]

Q2. Write a program which generates 500 integer values multiple of 2, store in an array. An integer search key (search a number) be entered. In case, the search key entered is not multiple of 6 the following message should display and ask again to enter the search key:

```
Search key is not multiple of 6
Enter the integer search key multiple of 6:
```

The following **recursive linear search function** should be called to find search key.

```
/* prototype of the function */
int search(int a[], int key, int l, int h); // l is low, h is high, key is search key
```

The function should return the index of the searched number, for example if search key is 66 then display the following message:

```
Found value in element 33
```

If the number is not found, the function should return -1 and display the following.

```
Value not found
```

[marks: 10]

Q3. Write a program which takes a character array and print its each character using 1). For-loop and 2). While-loop and also 3). as string. (Note: `fgets()` is allowed to use).

Write a **recursive function** `strRev` that takes a character array as an argument, prints it back to front i-e: in reverse, returns nothing and terminates at null character. [marks: 10]

If you type the following string (character array)

```
Hello I am here
```

Expected output will be the following:

```
The program started
Start typing:

Hello I am here

This is For Loop
Hello I am here

This is While Loop
Hello I am here

This is string
Hello I am here

The reverse string is;
ereh ma I olleH
```

Evaluation will be based on:

Correctness/structure of your program
Comments with proper explanation of each expression
You have 120 minutes to submit on Moodle:
Release time: 20:00 25-Nov-2022
Submission time: 22:00 25-Nov-2022