EF234101 Fundamental Programming (F)

Midterm Exam

Starting date:	16 October 2024
Deadline:	23 October 2024, 23:59 WIB. **Penalty: 0.15% of grade/minute of tardiness.**
Exam type:	Open, Individual Project
Send to:	MM Irfan Subakti <yifana@gmail.com> CC to Sarah Nurhasna Khairunnisa <sarahnurhasna@gmail.com>, Widian Sasi Disertasiani <widianogik@gmail.com>, Arya Gading Prinandika <gprinandika@gmail.com> & Adnan Abdullah Juan <adnanjuan06@gmail.com> with the subject: **EF234101_FunPro(F)_MID_StudentID_Name**
File type and format:	A **zip** file containing all of the **.c source files** & the **declaration**
Filename format:	**EF234101_FunPro(F)_MID_StudentID_Name.ZIP**

Instruction
Please do these steps as in the following.

1. Please create a program, namely `01_stats_[your_name].c`. At the beginning of the program, please write down the codes as in the following. Then continue the codes for max(), min(), sum(), average() and sDeviation(). You are not allowed to use the built-in function of C for max() and min(). You have to write these functions by yourself. [**25 points**]

```
#include <stdio.h>
#include <limits.h>
#include <math.h>

int max(int a[], unsigned int aSize);
int min(int a[], unsigned int aSize);
int sum(int a[], unsigned int aSize);
double average(int sum, unsigned int aSize);
double sDeviation(int a[], double mean, unsigned int aSize);

int main() {
    int i;
    int a[10] = {32, 27, 64, 18, 95, 14, 90, 70, 60, 37};
    printf("%s%13s\n", "Element", "Value");
    for (i = 0; i < 10; i++) {
         printf("%7d%13d\n", i, a[i]);
    }
    unsigned aSize = sizeof(a) / sizeof(a[0]);
    printf("Size of a = %d\n", aSize);
    printf("max(): %d\n", max(a, aSize));
    printf("min(): %d\n", min(a, aSize));
    int arraySum = sum(a, aSize);
    printf("sum(): %d\n", arraySum);
    double avg = average(arraySum, aSize);
    printf("average(): %.3f\n", avg);
    printf("sDeviation(): %.3f\n", sDeviation(a, avg, aSize));
```

```
    return 0;
}
int max(int a[], unsigned int aSize) {
    int i;
    int m = INT_MIN; /* INT_MIN from <limits.h> */
    for (i = 0; ... /* Please continue this function */
    ...
    ...
    ...
}
int min(int a[], unsigned int aSize) {
    int i;
    int m = INT_MAX; /* INT_MAX from <limits.h> */
    for (i = 0; ... /* Please continue this function */
    ...
    ...
    ...
}
int sum(int a[], unsigned int aSize) {
    int i;
    int s = 0;
    for (i = 0; ... /* Please continue this function */
    ...
    ...
    ...
}
double average(int sum, unsigned int aSize) {
    /* Please continue this function */
    ...
    ...
}
double sDeviation(int a[], double mean, unsigned int aSize) {
    int i;
    double s = 0.0;
    for (i = 0; ... /* Please continue this function */
    ...
    ...
    ...
}
```

The formula for the Standard Deviation (SD) can be seen in the following.

$$SD = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \bar{x})^2}{N}}$$

Where SD is the Standard Deviation of a sample, x_i is the observed values of the sample items, $\bar{x}$ is the mean value of this observation and N is the number of observations in the sample.

The output:

```
Element         Value
      0            32
      1            27
      2            64
      3            18
      4            95
      5            14
      6            90
      7            70
      8            60
      9            37
Size of a = 10
max(): 95
min(): 14
sum(): 507
average(): 50.700
sDeviation(): 27.673
```

2. Please create a program, namely `02_fibo_[your_name].c`, which be able to produce *n*-th Fibonacci number by (a) iterative (non-recursive) approach, and (b) recursive approach. For each of these approaches, i.e., (a) and (b), then you need to sum those resulting numbers up. [**25 points**]

   Input
   ```
   int number;
   ```

   The user needs to input this number by using `scanf()` function. Then you have to create a function to produce *n*-th Fibonacci number by iterative (non-recursive) and recursive methods. After it's done, please compare these functions, which one is the faster one? The iterative one or the recursive one? You may do the comparison with a number that is greater than 30.

   Output
   ```
   int iterativeResults;
   int sum;

   int recursiveResults;
   int sum;
   ```

   Input: the example
   ```
   Enter a number: 9
   ```

   Output: the example
   ```
   9th Fibonacci = 34
   Sum of Fibonacci until 9th term = 88
   ```

   Note
   Started by 0th for 0, the 9th Fibonacci number is: 0, 1, 1, 2, 3, 5, 8, 13, 21, **34**
   Sum of these Fibonacci numbers is 0 + 1 + 1 + 2 + 3 + 5 + 8 + 13 + 21 + 34 = **88**

The output:

```
Enter a number: 9

Iterative
9th Fibonacci = 34
Sum of Fibonacci until 9th term = 88
Iterative time: 98700 nanoseconds

Recursive
9th Fibonacci = 34
Sum of Fibonacci until 9th term = 88
Recursice time: 92600 nanoseconds

Recursive method is faster than iterative method by 6100 nanoseconds
```

The output:

```
Enter a number: 15

Iterative
15th Fibonacci = 610
Sum of Fibonacci until 15th term = 1596
Iterative time: 148900 nanoseconds

Recursive
15th Fibonacci = 610
Sum of Fibonacci until 15th term = 1596
Recursice time: 138700 nanoseconds

Recursive method is faster than iterative method by 10200 nanoseconds
```

The output:

```
Enter a number: 35

Iterative
35th Fibonacci = 9227465
Sum of Fibonacci until 35th term = 24157816
Iterative time: 113000 nanoseconds

Recursive
35th Fibonacci = 9227465
Sum of Fibonacci until 35th term = 24157816
Recursice time: 44599500 nanoseconds

Iterative method is faster than recursive method by 44486500 nanoseconds
```

3. Please create a program, namely `03_palindrome_[your_name].c`, where there is a string that will be defined as in the following. [**25 points**]

   ```
   char word[];
   ```

   Input
   - The string word.

Output

- "It is a palindrome" if it is a palindrome sentence.
- "It is not a palindrome" if it is not a palindrome sentence.
- The palindrome means that if we read the string from left or right, both will produce the same result. E.g., ABBA,
- Challenge: Use a method other than "just loop and branches", e.g., bitshift.
- Examples of a palindrome:
    - kayak
    - deified
    - rotator
    - repaper
    - deed
    - peep
    - wow
    - noon
    - civic
    - racecar
    - level
    - mom
    - amanaplanacanalpanama -> A man, a plan, a canal, Panama
    - borroworrob -> Borrow or rob
    - neveroddoreven -> Never odd or even
    - wepanicinapew -> We panic in a pew

The output:

```
Enter a word or sentence: kayak
It is a palindrome
```

The output:

```
Enter a word or sentence: julid
It is not a palindrome
```

The output:

```
Enter a word or sentence: amanaplanacanalpanama
It is a palindrome
```

4. Please create a program, namely `04_determinant.c`, which be able to calculate the determinant of the 3x3 matrix. [**25 points**]

    Output

    - The determinant of the matrix.
    - Invertibility of the matrix. If the determinant is 0, then the matrix cannot be inverted, otherwise, the matrix is invertible.

The output:

```
Enter the matrix:
1 3 2
-3 -1 -3
2 3 1
The matrix:
1 3 2
-3 -1 -3
2 3 1
Matrix determinant: -15
Matrix is invertible
```

5. Create a program to solve the following problem and name it `05_magic_herb_[your_name].c`. [**Challenge**]

The boy has safely returned to his home with his hard-worked spoils, a magical herb. That herb is said to be the key to cure his sickly mother. He was happy, he already imagined the face of his energetic mother, but he didn't know how to concoct the medicine from the herb. He only read it from a book, that his father left, that the herb was the key ingredient.

He was confused, it's so close yet so far. He reread his books, again and again, to check if he had missed something, without sleep nor eating for straight three days, and at long last he found a gibberish number hidden between the pages. That's the magic word to activate the true nature of the book! The boy yells happily reminiscing the figure of his long-lost father. As if remembering something the boy went to his father's study and took a piece of paper.

The number is
`{94,222,221,186,181,208,207,110,187,185,17,212,115,215,100}`

The method to decipher the gibberish number that is written on that paper

You must proceed the number one by one from the rightmost number, i.e., 100, to the leftmost one, i.e., 94. It means the deciphering procedure is started from the right to the left. The rightmost number, 100, is the first number (n_0) to decipher the gibberish number. By considering the next number (remember, we move from the right to the left), i.e., 215, then we are going to find the **designated number (n_i) to be deciphered**. Finding that **designated number** can be done by performing one of the four operations, as in the following.

Operation 1:

$$n_{i+1} = n_{i+1} + n_i$$

Operation 2:

$$n_{i+1} = n_{i+1} - n_i$$

Operation 3:

$$n_{i+1} = (n_{i+1} * 2) - n_i$$

Operation 4:

$$n_{i+1} = (n_{i+1} - 3) * 2 + n_i$$

Which of those operations the **designated number** is whether n_{i+1} is between 65 (inclusive) and 122 (inclusive) or not? In other words, whether $65 \leq n_{i+1} \leq 122$ (between 'A' and 'z') or not?

Example. Remember, the deciphering process starts from the right to the left.

1st iteration

i = 0 → n_0 = 100, n_1 = 215

Operation 1:

n_1 = 215 + 100 = 315 (false, it's not the **designated number**)

Operation 2:

n_1 = 215 - 100 = **115** (true, the number is between 65 – 122, so **115** is the **designated number**. The ASCII code for **115** is **s**, so **s** is the last **deciphered character** of the magic word)

Operation 3:

n_1 = (215 * 2) - 100 = 330 (false, it's not the **designated number**)

Operation 4:

n_1 = (215 – 3) * 2 + 100 = 524 (false, it's not the **designated number**)

We got the **designated number**, i.e., **115**, and the current number now is below.

{94,222,221,186,181,208,207,110,187,185,17,212,115,**115**,**100**}

2nd iteration

i = 1 → n_1 = 115, n_2 = 115

Operation 1:

n_2 = 115 + 115 = 230 (false, it's not the **designated number**)

Operation 2:

n_2 = 115 - 115 = 0 (false, it's not the **designated number**)

Operation 3:

n_2 = (115 * 2) - 115 = 115 (true, the number is between 65 – 122, so **115** is the **designated number**. The ASCII code for **115** is **s**, so **s** is the second last **deciphered character** of the magic word)

Operation 4:

n_2 = (115 – 3) * 2 + 115 = 339 (false, it's not the **designated number**)

We got the **designated number**, i.e., **115**, and the current number now is below.

{94,222,221,186,181,208,207,110,187,185,17,212,**115**,**115**,**100**}

3rd iteration

i = 2 → n_2 = 115, n_3 = 212

Operation 1:

n_3 = 212 + 115 = 327 (false, it's not the **designated number**)

Operation 2:

n_3 = 212 - 115 = **97** (true, the number is between 65 – 122, so **97** is the **designated number**. The ASCII code for **97** is **a**, so **a** is the third last **deciphered character** of the magic word)

Operation 3:

n_3 = (212 * 2) - 115 = 309 (false, it's not the **designated number**)

Operation 4:

n_3 = (212 – 3) * 2 + 115 = 533 (false, it's not the **designated number**)

We got the **designated number**, i.e., **97**, and the current number now is below.

`{94,222,221,186,181,208,207,110,187,185,17,97,115,115,100}`

…… and so, on

Solve it by using the recursive function. Note: the last number (the rightmost one, n_0 = 100) produced the last **deciphered character 's'** of the **designated number '115'** from the result of operation 2 in the 1st iteration. It means we got the magic word " … s". Then by a similar procedure, we got the magic word " … ass" after the 2nd iteration (n_1 = 215 by operation 3) and the 3rd iteration (n_2 = 115 by operation 2), respectively.

Output

Print the ASCII character of the **designated number**. Hint: you can use `printf("%c\n", number)` to print the ASCII character of the designated number.

Example output:

```
?
?
?
.
.
.
.
.
.
.
.
.
a
s
s
```

6. To avoid plagiarism/cheating, every student needs to pledge and declare, then she/he must submit her/his **signed pledge and declaration** as in the following. Failing to do so will result in getting a 0 (zero) grade. Attach the **scanned**/**photo** of your *declaration* in your report.

   "By the name of Allah (God) Almighty, herewith I pledge and truly declare that I have solved the midterm exam by myself, didn't do any cheating by any means, didn't do any plagiarism, and didn't accept anybody's help by any means. I am going to accept all of the consequences by any means if it has proven that I have done any cheating and/or plagiarism."

   [Place, e.g., Surabaya], [date, e.g., 23 October 2024]

   <Signed>

   [Full name, e.g., Arianti Purbasari]
   [StudentID, e.g., 05112440000xxxx]

7. ZIP the files of `01_stats_[your_name].c`, `02_fibo_[your_name].c`, `03_palindrome_[your_name].c`, `04_determinant_[your_name].c`, `05_magic_herb_[your_name].c` and your declaration (e.g., `Declaration.PDF`) into 1 (one) only .ZIP file, namely `EF234101_FunPro(F)_MID_StudentID_Name.ZIP`. Send this .ZIP file to `yifana@gmail.com` and CC it to the TA's emails.

8. Have a wonderful day! Good luck! 😊