

TI184411 Algorithms & Computer Programming (English)

Midterm Exam

Starting date: 9 March 2019
 Deadline: 16 March 2019, 23:59 WIB.
 Exam type: Open
 Send to: MM Irfan Subakti <yifana@gmail.com>
 CC to Irham Muhammad Fadhil <irham.mfadhil@gmail.com>
 with the subject: TI184411_ALPRO_MID_NRP_Name

File type and format: A zip file containing all of the .c source files & the declaration
 Filename format: TI184411_ALPRO_MID_NRP_Name.ZIP

Instruction

Please do these steps as in the following.

1. Please create a program, namely 01_stats.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 these observation and N is the number of observations in the sample.

2. Please create a program, namely `02_fibo.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 resulted 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 the number that greater than 30.

Output

```
int iterativeResults;
int sum;

int recursiveResults;
int sum;
```

Input: the example

9

Output: the example

```
9th Fibonacci = 34
Sum of Fibonacci until 9th term = 88
```

Note

The Fibonacci numbers until 9th term are: 1, 1, 2, 3, 5, 8, 13, 21, **34**
Sum of these Fibonacci numbers is $1 + 1 + 2 + 3 + 5 + 8 + 13 + 21 + 34 = \mathbf{88}$

3. Please create a program, namely `03_string.c`, where two strings will be defined as in the following. [25 points]

```
char word1[] = "Happy New Year";
char word2[] = "Happy Holidays";
```

Then concatenate these two strings, calculate the length of the concatenated string. Finally, reverse that concatenated string.

Output

- The result of the concatenated string.
 - The length of the concatenated string.
 - The reverse of the concatenated string. E.g., if the concatenated string is "Hello", then the reserved of the concatenated string is "olleH".
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.

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. Failed to do so will be resulted in getting 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 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 been done any cheating and/or plagiarism.”

[Place, e.g., Surabaya], [date, e.g., 16 March 2019]

<Signed>

[Full name, e.g., Mukidi Mukiyo]

[NRP, e.g., 05111340000xxx]

7. ZIP the files of `01_stats.c`, `02_fibo.c`, `03_string.c`, `04_determinant.c` and your declaration (e.g., `Declaration.PDF`) into 1 (one) only .ZIP file, namely `TI184411_ALPRO_MID_NRP_Name.ZIP`. Send this .ZIP file to `yifana@gmail.com` and CC-ed to `irham.mfadhil@gmail.com`.

8. Have a great day! Good luck! 😊