

CSE-016 Programming Lab Assignment № 5

10/05/2024

Youssef Ahmed Samy Kassem

ID 9545 – Group 3 – Lab 1

SSP – Faculty of Engineering, Alexandria University

Solutions begin from the second page.

1 Problems

1.1 Problem (1)

Write a program that reads N integers then finds and prints the average and the variance of the entered values.

1.2 Problem (2)

Write a program to check whether a given number is a perfect number or not. [Perfect number is a positive number which sum of all its positive divisors excluding itself is equal to that number.]

Example 1:

Enter a number: 56

The positive divisors: 1 , 2 , 4 ,
7 , 8 , 14 , 28 ,

The sum of the divisors is 64
56 is not a perfect number

Example 2:

Enter a number: 28

The positive divisors: 1 , 2 , 4 ,
7 , 14 ,

The sum of the divisors is 28
28 is a perfect number

Example 3:

Enter a number: 496

The positive divisors: 1 , 2 , 4 ,
8 , 16 , 31 , 62 , 124 , 248 ,

The sum of the divisors is 496
496 is a perfect number

Contents

1 Problems	1	2.2 Solution to Problem (2)	3
1.1 Problem (1)	1	2.2.1 Source Code	3
1.2 Problem (2)	1	2.2.2 Outcome	4
2 Solutions	2	2.3 Evidence of Work (Screenshots) .	5
2.1 Solution to Problem (1)	2	2.4 Specifications	6
2.1.1 Source Code	2		
2.1.2 Outcome	2		
		3 Licenses	6

2 Solutions

2.1 Solution to Problem (1)

2.1.1 Source Code

Program's C code | Line numbers for readability

```
1  #include <stdio.h>
2  #include <math.h>
3  void main(){
4      int n = 1, i, total = 0, j;
5      double avg, variance, summation = 0;
6      do {
7          if (n < 1) printf("Value error. Only positive integers allowed.\n");
8          printf("Enter the sample size: ");
9          scanf("%d", &n);
10     } while (n < 1);
11     int sample[n] = {}, width = floor(log10(n)) + 1;
12     for (i = 1; i <= n; i++) {
13         printf("Enter number (%*d/%d): ", width, i, n);
14         scanf("%d", &sample[i-1]);
15         total += sample[i-1];
16     }
17     avg = (double) total/n;
18     for (j = 0; j < n; j++){
19         summation += pow(sample[j] - avg, 2);
20     }
21     variance = (double) summation / (n - 1);
22     printf("Average: %f\nVariance: %f", avg, variance);
23 }
```

2.1.2 Outcome

Program's output to console in plaintext – 25 as sample size

```
Enter the sample size: 10
Enter number ( 1/10): 1
Enter number ( 2/10): 2
Enter number ( 3/10): 3
Enter number ( 4/10): 4
Enter number ( 5/10): 5
Enter number ( 6/10): 6
Enter number ( 7/10): 7
Enter number ( 8/10): 8
Enter number ( 9/10): 9
Enter number (10/10): 10
Average: 5.500000
Variance: 9.166667
```

2.2 Solution to Problem (2)

2.2.1 Source Code

Program's C code – uses number theory to determine if number is perfect

```
1  #include <stdio.h>
2  void main(){
3      unsigned long long input = 1, buffer, sum = 0;
4      int multiplicity_of_2 = 0, is_perfect = 0;
5      do {
6          if (input < 1) printf("Value error. Only positive integers allowed.\n");
7          printf("Enter a number: ");
8          scanf("%lld", &input);
9      } while (input < 1);
10     buffer = input;
11     do {
12         if (buffer % 2 != 0) {
13             if (multiplicity_of_2 > 0){
14                 unsigned long long even_pair = input / buffer;
15                 if (2*even_pair - 1 == buffer) is_perfect = 1;
16             }
17             break;
18         }
19         buffer /= 2; multiplicity_of_2++;
20     } while (buffer > 1);
21     printf("The positive divisors: "); /* Spacing as intended in lab note */
22     if (is_perfect) {
23         int i, k;
24         unsigned long long j = 1;
25         for (k = 1; k >= 0; k--) {
26             for (i = 0; i < multiplicity_of_2 + k; i++) {
27                 printf("%lld , ", j);
28                 j *= 2;
29             }
30             j = buffer;
31         }
32     } else {
33         unsigned long long i;
34         for (i = 1; i < input; i++){
35             if (input % i != 0) continue;
36             sum += i;
37             printf("%lld , ", i);
38         }
39     }
40     printf("\nThe sum of divisors is %lld\n", (is_perfect ? input : sum));
41     printf("%lld is %s perfect number", input, (is_perfect ? "a" : "not a"));
42 }
```

2.2.2 Outcome

This program *first* checks if the number is perfect, before displaying all its proper factors and their sum. The output is exactly the same, but the result is much less computationally expensive code and support for larger inputs without huge delays. Uses *Mersenne Primes* (line 15).

Program's output to console in plaintext – 56 as input

```
Enter a number: 56
The positive divisors: 1 , 2 , 4 , 7 , 8 , 14 , 28 ,
The sum of divisors is 64
56 is not a perfect number
```

Program's output to console in plaintext – 28 as input

```
Enter a number: 28
The positive divisors: 1 , 2 , 4 , 7 , 14 ,
The sum of divisors is 28
28 is a perfect number
```

Program's output to console in plaintext – 496 as input

```
Enter a number: 496
The positive divisors: 1 , 2 , 4 , 8 , 16 , 31 , 62 , 124 , 248 ,
The sum of divisors is 496
496 is a perfect number
```

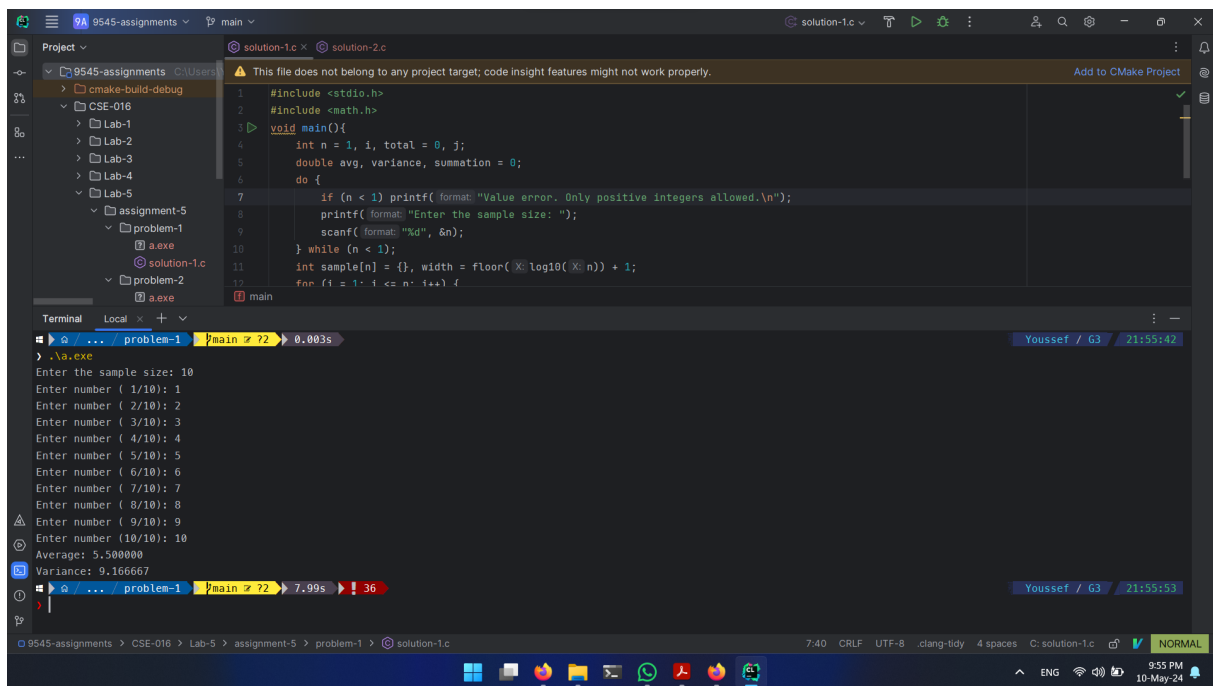
Program's output to console in plaintext – 33550336 as input (perfect)

```
Enter a number: 33550336
The positive divisors: 1 , 2 , 4 , 8 , 16 , 32 , 64 , 128 , 256 , 512 , 1024 , 2048 ,
  4096 , 8191 , 16382 , 32764 , 65528 , 131056 , 262112 , 524224 , 1048448 , 2096896 ,
  4193792 , 8387584 , 16775168 ,
The sum of divisors is 33550336
33550336 is a perfect number
```

Program's output to console in plaintext – 33550330 as input (not perfect)

```
Enter a number: 33550330
The positive divisors: 1 , 2 , 5 , 10 , 11 , 22 , 23 , 46 , 55 , 89 , 110 , 115 , 149
  , 178 , 230 , 253 , 298 , 445 , 506 , 745 , 890 , 979 , 1265 , 1490 , 1639 , 1958
  , 2047 , 2530 , 3278 , 3427 , 4094 , 4895 , 6854 , 8195 , 9790 , 10235 , 13261 , 16390
  , 17135 , 20470 , 22517 , 26522 , 34270 , 37697 , 45034 , 66305 , 75394 , 112585
  , 132610 , 145871 , 188485 , 225170 , 291742 , 305003 , 376970 , 610006 , 729355
  , 1458710 , 1525015 , 3050030 , 3355033 , 6710066 , 16775165 ,
The sum of divisors is 36433670
33550330 is not a perfect number
```

2.3 Evidence of Work (Screenshots)



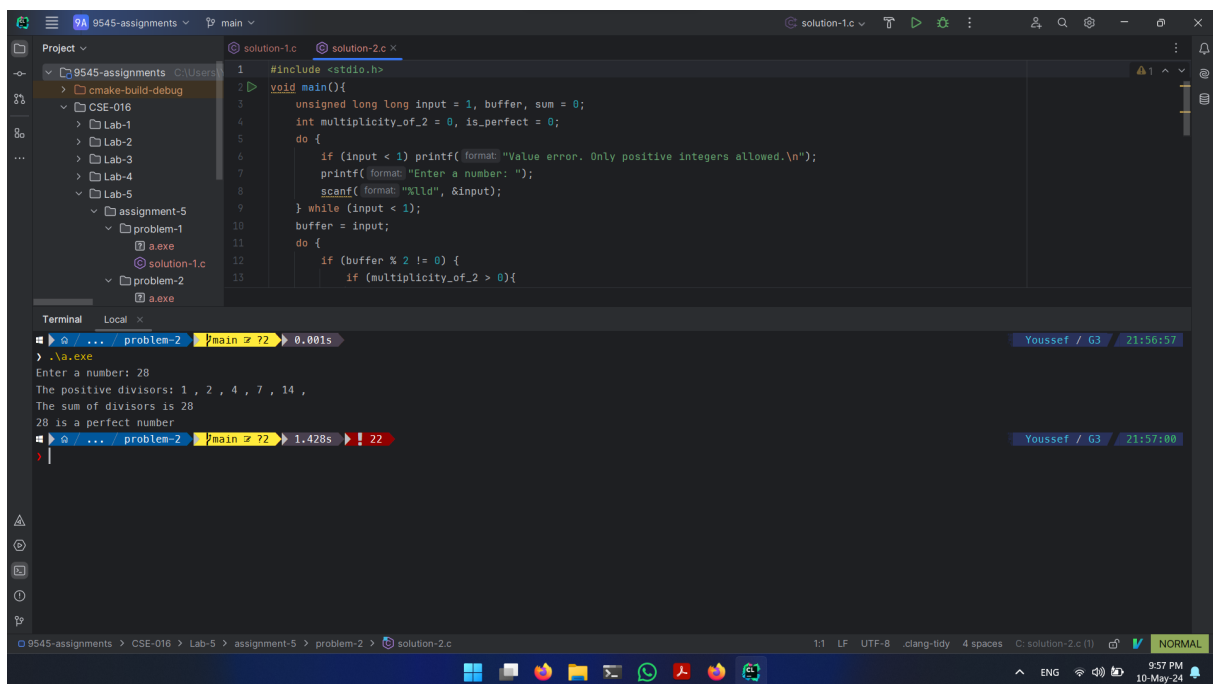
The screenshot shows the CLion IDE interface. The project is '9545-assignments' and the current file is 'solution-1.c'. The code in the editor is as follows:

```
1 #include <stdio.h>
2 #include <math.h>
3 void main()
4 {
5     int n = 1, i, total = 0, j;
6     double avg, variance, summation = 0;
7     do {
8         if (n < 1) printf( format: "Value error. Only positive integers allowed.\n");
9         printf( format: "Enter the sample size: ");
10        scanf( format: "%d", &n);
11    } while (n < 1);
12    int sample[n] = {}, width = floor( (X log10( X n) ) + 1;
13    for (i = 1; i <= n; i++) {
```

The terminal window shows the execution of the program. It prompts the user to enter the sample size (10) and then enters numbers 1 through 10. The output shows the average (5.500000) and variance (9.166667).

```
Youssef / G3 21:55:42
> .\a.exe
Enter the sample size: 10
Enter number ( 1/10): 1
Enter number ( 2/10): 2
Enter number ( 3/10): 3
Enter number ( 4/10): 4
Enter number ( 5/10): 5
Enter number ( 6/10): 6
Enter number ( 7/10): 7
Enter number ( 8/10): 8
Enter number ( 9/10): 9
Enter number (10/10): 10
Average: 5.500000
Variance: 9.166667
Youssef / G3 21:55:53
```

Figure 1: Desktop screenshot of Problem (1)'s code in CLion



The screenshot shows the CLion IDE interface. The project is '9545-assignments' and the current file is 'solution-2.c'. The code in the editor is as follows:

```
1 #include <stdio.h>
2 void main()
3 {
4     unsigned long long input = 1, buffer, sum = 0;
5     int multiplicity_of_2 = 0, is_perfect = 0;
6     do {
7         if (input < 1) printf( format: "Value error. Only positive integers allowed.\n");
8         printf( format: "Enter a number: ");
9         scanf( format: "%lld", &input);
10    } while (input < 1);
11    buffer = input;
12    do {
13        if (buffer % 2 != 0) {
14            if (multiplicity_of_2 > 0){
```

The terminal window shows the execution of the program. It prompts the user to enter a number (28). The output shows the positive divisors (1, 2, 4, 7, 14) and the sum of divisors (28). It also indicates that 28 is a perfect number.

```
Youssef / G3 21:56:57
> .\a.exe
Enter a number: 28
The positive divisors: 1 , 2 , 4 , 7 , 14 ,
The sum of divisors is 28
28 is a perfect number
Youssef / G3 21:57:00
```

Figure 2: Desktop screenshot of Problem (2)'s code in CLion

2.4 Specifications

- **Libraries:**
 - `stdio.h`
 - `math.h` (problem 1)
- **Compiler:** GNU C Compiler (`gcc`) version 13.2.0 (x86_64-posix-seh-rev1)
- **C Standard Compatibility**

P#	C89/C90	C99	C11	C17	C23
1	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓

- **Supported Platforms:** OS: (any), architecture: (any)
- **Tested On:** Windows 11 64-bit

3 Licenses

This document, my additions to its L^AT_EX source code, the software included and its C source code all come without warranty and are all subject to the BSD 3-Clause Open Source License: <https://opensource.org/license/bsd-3-clause>.

COPYRIGHT © 2024, Youssef Ahmed Samy