



Homework # 3

**01286131 Object-Oriented Programming
Software Engineering Program,
Department of Computer Engineering,
School of Engineering, KMITL**

By

65011277 Chanasorn Howattanakulphong

Practice with the C++ Standard Library

Note: You **must** structure every programs into appropriate functions responsible for reading data input, calculating values, and writing output and also choose appropriate data types to store values.

1. Given the following example input data file that contains two sequences separated by a semicolon:

data1.txt

3 1 2 7 5 6 13; 20 10 5 40

data2.txt

21 5 42 12; 13 1 3 2 7 5

1.1) Write a program to read two sequences data from a file and combine them by interleaving the data, appending the rest from the longer sequence.

Results for data1.txt

3 20 1 10 2 5 7 40 5 6 13

Results for data2.txt

21 13 5 1 42 3 12 2 7 5

1.2) Write a program to read two sequences data from a file and calculate the sum of product from them.

Calculations from data1.txt

$(3 * 20) + (1 * 10) + (2 * 5) + (7 * 40)$
= 360

Calculations from data2.txt

$(21 * 13) + (5 * 1) + (42 * 3) + (12 * 2)$
= 428

1.3) Write a program to read two sequences data from a file and generate the cartesian product from them.

Results from data1.txt

(3, 20), (3, 10), (3, 5), (3, 40),
(1, 20), (1, 10), (1, 5), (1, 40),
(2, 20), (2, 10), (2, 5), (2, 40),
(7, 20), (7, 10), (7, 5), (7, 40), (5, 2), (5, 7), (5, 5), (5, 20), (5, 10),
(5, 5), (4, 40), (42, 13), (42, 1), (42, 3),
(6, 20), (6, 10), (6, 5), (6, 40),
(13, 20), (13, 10), (13, 5), (13, 40)

Results from data2.txt

(21, 13), (21, 1), (21, 3),
(21, 2), (21, 7), (21, 5),
(5, 13), (5, 1), (5, 3),
(42, 2), (42, 7), (42, 5),
(12, 13), (12, 1), (12, 3),
(12, 2), (12, 7), (12, 5)

<data1.txt>

Interleave :
3 20 1 10 2 5 7 40 5 6 13

Sum of products :
360

Cartesian product :
(3, 20), (3, 10), (3, 5), (3, 40), (1, 20), (1, 10), (1, 5), (1, 40), (2, 20), (2, 10), (2, 5), (2, 40), (7, 20), (7, 10), (7, 5), (7, 40), (5, 20), (5, 10),
(5, 5), (5, 40), (6, 20), (6, 10), (6, 5), (6, 40), (13, 20), (13, 10), (13, 5), (13, 40),

PS D:\Main\Work\KMIL\Yr1 Sem2\OOP\HW\HW_3>

<data2.txt>

Interleave :
21 13 5 1 42 3 12 2 7 5

Sum of products :
428

Cartesian product :
(21, 13), (21, 1), (21, 3), (21, 2), (21, 7), (21, 5), (5, 13), (5, 1), (5, 3), (5, 2), (5, 7), (5, 5), (42, 13), (42, 1), (42, 3), (42, 2), (42, 7), (42, 5),
(12, 13), (12, 1), (12, 3), (12, 2), (12, 7), (12, 5),

2. Given the following example input data file that contains sequences separated by a semicolon:

Example data input

3.5 1.2 2.4; 4.2 2.7; 3 7 2

2.1) Write a program to read the data file and calculate an average for each sequence. *Example output for 2.1)*

2.36 6.9 4

2.2) Write a program to read the data file and generate the cartesian product from them. *Example output for 2.2)*

(3.5, 4.2, 3), (3.5, 4.2, 7), (3.5, 4.2, 2),
(3.5, 2.7, 3), (3.5, 2.7, 7), (3.5, 2.7, 2),
(1.2, 4.2, 3), (1.2, 4.2, 7), (1.2, 4.2, 2),
(1.2, 2.7, 3), (1.2, 2.7, 7), (1.2, 2.7, 2),
(2.4, 4.2, 3), (2.4, 4.2, 7), (2.4, 4.2, 2),
(2.4, 2.7, 3), (2.4, 2.7, 7), (2.4, 2.7, 2)

```
gawk -F ';' '{n1=length($1);n2=length($2);n3=length($3);
<data3.txt>

vec1 :
3.5
1.2
2.4
vec2 :
4.2
2.7
vec3 :
3
7
2
Average of v1: 2.36667
Average of v2: 3.45
Average of v3: 4
```

```
Cartesian Product of v1, v2, v3:
(3.5, 4.2, 3), (3.5, 4.2, 7), (3.5, 4.2, 2), (3.5, 2.7, 3), (3.5, 2.7, 7), (3.5, 2.7, 2), (1.2, 4.2, 3), (1.2, 4.2, 7), (1.2, 4.2, 2), (1.2, 2.7, 3), (1.2, 2.7, 7), (1.2, 2.7, 2), (2.4, 4.2, 3), (2.4, 4.2, 7), (2.4, 4.2, 2), (2.4, 2.7, 3), (2.4, 2.7, 7), (2.4, 2.7, 2),
```

3. Write a program to split words into a group of words with enclosing "... " pairs, and a group without, and then generate pairs of words from them:

Example data input

Example results

Red *Green* Blue Cyan *Black* White

(*Green*, Red), (*Green*, Blue),
(*Green*, Cyan), (*Green*, White),
(*Black*, Red), (*Black*, Blue),
(*Black*, Cyan), (*Black*, White)

```
<data4.txt>

With style: *Green* *Black*
Without style: Red Blue Cyan White
Cartesian product: (*Green*, Red), (*Green*, Blue), (*Green*, Cyan), (*Green*, White), (*Black*, Red), (*Black*, Blue), (*Black*, Cyan), (*Black*, White),
```