

1. The Fibonacci sequence is a sequence that meets the following condition.

$$F_1 = F_2 = 1$$

$$F_n = F_{n-1} + F_{n-2} \quad (n \in \{3, 4, \dots\})$$

Implement Fibonacci sequence that works for all Ns in $0 < N < 51$.

Program gets an integer N and outputs a Number of Nth Fibonacci.(1pt)

Example) Number: 50
12586269025

Number: 4
3

2. Implement a Program that inputs the size of Matrix and Adds all in a given direction.
The Array is a Square matrix of size N($0 < N < 10$) that fills all with 1 and The Direction is obtained by integer.
Program's Sequence is as follows.(1pt)

- 1) Receives integer N that means Matrix Size.
- 2) Output Square matrix of size N that filled all with 1.
- 3) Receives integer M that means direction.
- 4) Output Square matrix of size N that added in the direction entered

However, it is prohibited to implement the method of calculating and substituting the final destination using that all terms in the matrix are 1.(ex. Prohibit $\text{Array}[0][0] = 1 * N$)

Command List

0 : Up 1 : Down 2 : Right 3 : Left 4 : Down Left 5 : Up Right

Example)

Array Size($0 < N < 10$):3 1 1 1 1 1 1 1 1 1 Shift All Direction:0 3 3 3 0 0 0 0 0 0	Array Size($0 < N < 10$):3 1 1 1 1 1 1 1 1 1 Shift All Direction:1 0 0 0 0 0 0 3 3 3	Array Size($0 < N < 10$):3 1 1 1 1 1 1 1 1 1 Shift All Direction:2 0 0 3 0 0 3 0 0 3
<UP>	<Down>	<Right>
Array Size($0 < N < 10$):3 1 1 1 1 1 1 1 1 1 Shift All Direction:3 3 0 0 3 0 0 3 0 0	Array Size($0 < N < 10$):3 1 1 1 1 1 1 1 1 1 Shift All Direction:4 1 0 0 2 0 0 3 2 1	Array Size($0 < N < 10$):3 1 1 1 1 1 1 1 1 1 Shift All Direction:5 1 2 3 0 0 2 0 0 1
<Left>	<Down Left>	<Up Right>

3. Implement Program that Receives N that size of a matrix and sort that matrix.($0 < N < 11$)
The matrix is a square matrix of size N and fills all with random numbers. random numbers are an integer greater than 0 and less than 100.(1pt)

Condition

- Matrix must have different values for each execute
- Sorting is in ascending order, and $(0,0) \leq (0,1) \leq (0,2) \leq \dots \leq (1,0) \leq (1,1) \dots \leq (N-1,N-1)$ order.
- Implement and use sorting function directly

Example)

Array Size($0 < N < 11$) : 5				
59	33	29	15	64
5	89	87	84	71
68	8	76	55	34
41	57	99	46	39
52	3	81	65	42
3	5	8	15	29
33	34	39	41	42
46	52	55	57	59
64	65	68	71	76
81	84	87	89	99

4. Implement a Program that removes string except english and prints that string.
The program Receives the string and outputs the converted string.
All english characters are converted to lowercase letters and all characters except english are removed. assumed that spacing is not entered.(1pt)

Example)

iHateApple	i_Love_app!le@@
ihateapple	iloveapple

5. Implement a Program that compresses repeated characters. The program expresses repeated characters with character + number of repeated. for example, aaaa is a4
The program receives string and outputs converted string. (Exclude 2byte character Input) (1pt)

Role

- Compress all characters that are repeated more than two times.
- The character that are compressed are expressed as the corresponding characters + number of repeated

Example)

aaaacbbdddee	aaa !!!tcccaabbb
a4c2bd3e2	a3 4!3tc3a2b3

6. Checksum is a method of verifying the integrity of the transmitted data. Implement Simple Checksum Simulator.

Simple Checksum Simulator consists of three functions: Sender, Transmission_Process, Receiver.

a. Sender

- input : -
- output : -
- Description : get 4-digit character data
Calculate data's checksum.
Print data + checksum
Send data + checksum to Transmission_Process

b. Transmission_Process

- input : const char* data
- output : -
- Description : Generate Error (40%)
Send data + checksum to Receiver

c. Receiver

- input : const char* data
- output : -
- Description : Data Verification
Print Received Data & result(false : Error Occur, True : Error not Occur)

Checksum's Rule is follows

a. Sender

- a) Make a checksum number. checksum number is the first digit of the sum of each data's digit
- b) Attach checksum number at data's edge

ex) Data : 4752

Checksum number : $(4+7+5+2 = 18) \Rightarrow 8$

transmission data : data + 8

b. Receiver

- a) make a Received data's checksum number again.
- b) If the made checksum number equal to Received checksum number, Error not Occurred else Error is Occurred.

ex) Received Data : 47529

Checksum number : $(4+7+5+2 = 18) \Rightarrow 8$

8 is not equal to 9 -> Error!

Transmission_Process's Error Occurred Percent is 40%.An Error is a random data change of one of five digit data.

Simple Checksum Simulator should receive a four-digit number and should print send data and received data and verified results.(1pt)

Example)

```
Data : 4741
Send Data: 47416
Received Data: 47416
6==6      Error_is not Occured!
```

```
Data : 3212
Send Data: 32128
Received Data: 32228
8!=9      Error!
```

7. Implement a program that validates formulas with dimensions 1 or less. Only x is entered for the variable. The Program follows the following condition.
- Exclude all Inputs except (,),+,-,/,*,Number,x.
 - All parentheses must be paired.
 - Don't skip *.
 - The left and right sides of +,-,/,* must have a number, x, or valid parenthesis.

The program receives an expression of the dimensions 1 or less, and outputs 'faulty' if the expression is defective, and 'integrity' if it is flawless.(2pt)

Example) <You do not need to follow the example output method, only result (faulty,integrity) is Essential>

```
formula: x+6*(7+2/(2+x))
2+x
7+2/(2+x)
x+6*(7+2/(2+x))
(x+6*(7+2/(2+x)))
integrity
```

```
formula: (x+2*(25+x)
25+x
x+2*(25+x)
(x+2*(25+x))
fualty
```

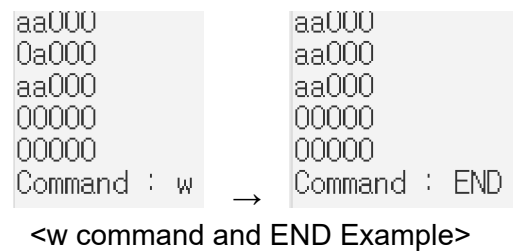
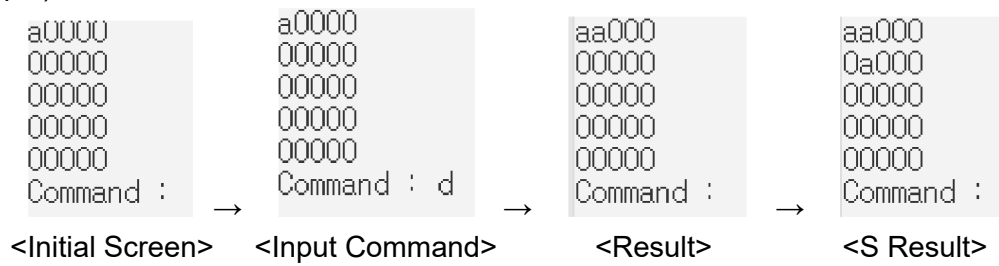
8. Implement a Program that moves the matrix and fills the matrix with 'a' Matrix is a square matrix of size 5 that is filled with '0'. Program receives command by user and fills the matrix with a as command. If all sides of the current location are filled with 'a', Program outputs 'END' and be Finished. Command list is as follows

a : Move Left
d : Mode Right

w : Move Up
s : Move Down

If Destination was filled 'a' already, the command is canceled. Initialize and redraw the console window whenever the command is performed.(2pt)

Example)



문의 : 박우혁조교(pwh7514@gmail.com, 새빛관 710호)

Due Date : 3월 26일(금) 23:59

Delay Date : 3월 27일(토) 11:59