

- Username : class-1
 - Password :
-
- Check VS 2019 whether can use
 - We will start our course in 18:30
 - we will start demonstrate the exercises at 19:15.
 - Do not use scanf_s
 - Please make sure the TA has recorded your exercise score [here](#) before leaving.

Array and Dynamic Array

Array: memory is allocated at compile time having a fixed size of it. You cannot alter or update the size of this array.

How should the following code be modified?

```
int n;  
scanf("%d", &n);  
int arr[n];
```

Solution 1 - const int

If there is a limit on the size of the array, you can use this approach.

```
const int arr_size = 105;  
scanf("%d", &n);  
int arr[arr_size];      // equal to int arr[105];
```

For example, there is a integer n ($0 \leq n \leq 100$) in the first line and there are n integers in the second line.

What if the array size is large or unlimited?

Solution 2 - Dynamic Array

If the array size is large or unlimited, you can use dynamic array.

```
int n;  
scanf("%d", &n);  
int* arr = NULL;  
arr = (int*)calloc(n, sizeof(int));
```

Scanf - Read char and string

Read string:

5
abcde

```
int n;  
char in_char[10];  
scanf("%d", &n);  
for (int i = 0; i < n; i++)  
    scanf("\n%c", &in_char[i]);
```

```
int n;  
char in_string[10];  
scanf("%d", &n);  
scanf("%s", &in_string);
```

Read sentence:

2
Hello world!
I love coding!

```
int n;  
char in_char[20], in_string[5][20];  
scanf("%d\n", &n);  
  
for(int i=0;i<n;i++)  
    scanf("\n%[^\\n]s", &in_string[i])
```

Homework Bonus

- There will be two bonus homework, each contributing one additional point to the total assignment score (with a maximum homework score of 20 points).
- The other one will be announced as soon as possible.
- These two bonus homework will not be on the Online Judge.

File I/O

- Introduction to Computers and Programming
- Week11 TA Course 2023/11/21

Outline

- Open a file
- Process a file
 - Read from the file
 - Write to the file
 - Reposition stream position indicator
- Close a file
- Exercises

Open a file



fopen()

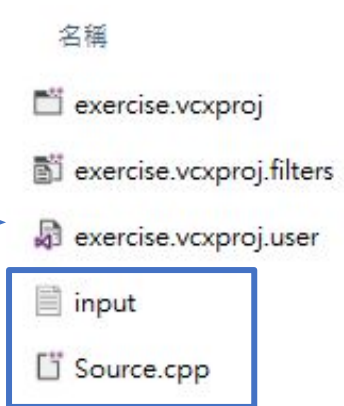
- FILE *fptr
 - Declare a variable named “fptr”.
 - The pointer that points to the location of a file.

• `fptr = fopen(“filename.txt”, “mode”)`

The file you want to open.

1. Default is the same path to cpp file
2. You can also specify the path of the file

The mode you want to do to the file.



- The return is the location of “filename.txt”

fopen()

• `fptr = fopen("filename.txt", "mode")`



"r"	read: Open file for input operations. The file must exist.
"w"	write: Create an empty file for output operations. If a file with the same name already exists, its contents are discarded and the file is treated as a new empty file.
"a"	append: Open file for output at the end of a file. Output operations always write data at the end of the file, expanding it. Repositioning operations (fseek , fsetpos , rewind) are ignored. The file is created if it does not exist.
"r+"	read/update: Open a file for update (both for input and output). The file must exist.
"w+"	write/update: Create an empty file and open it for update (both for input and output). If a file with the same name already exists its contents are discarded and the file is treated as a new empty file.
"a+"	append/update: Open a file for update (both for input and output) with all output operations writing data at the end of the file. Repositioning operations (fseek , fsetpos , rewind) affects the next input operations, but output operations move the position back to the end of file. The file is created if it does not exist.

Process a file



Read from the file

- `fscanf()` **`fscanf(file pointer, "datatype", data variable)`**
Reads data from the file and stores them into data variables.
Ex. `fscanf(fp, "%f", &x);`
- `fgetc()` **`variable = fgetc(file pointer)`**
Reads a character currently pointed by the file pointer and return the character.
Ex. `char C = fgetc(fp);`
- `fgets()` **`fgets(string, num, file pointer)`**
Reads characters from the file and stores them as a C string until (num-1) characters have been read.
Ex. `fgets(str, 60, fp);`

Example

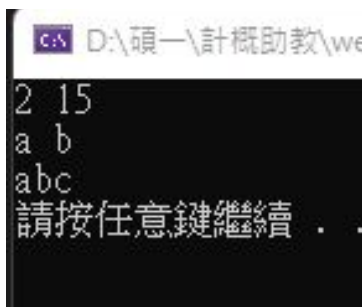


```
int main() {
    FILE* fptr;
    // Open the file
    fptr = fopen("input.txt", "r");

    int num1 = 0, num2 = 0;
    char c1, c2;
    char str[5];

    // Check whether the file is opened
    if (fptr != NULL) {
        // Read two numbers from the file and store them into the variables
        fscanf(fptr, "%d %d", &num1, &num2);
        printf("%d %d\n", num1, num2);
        // Read one character from the file
        c1 = fgetc(fptr);
        c2 = fgetc(fptr);
        printf("%c %c\n", c1, c2);
        // Read a string from the file
        fgets(str, 5, fptr);
        printf("%s\n", str);
    }
    else {
        // If fail to open the file, print the error message
        printf("Fail to open the 'input.txt'.");
    }

    // Close the file
    fclose(fptr);
}
```



Write to the file

fprintf(file pointer, "datatype", data variable)

fprintf() Writes the C string pointed by format to the file.
Ex. `fprintf(fp, "float number is %f", x);`

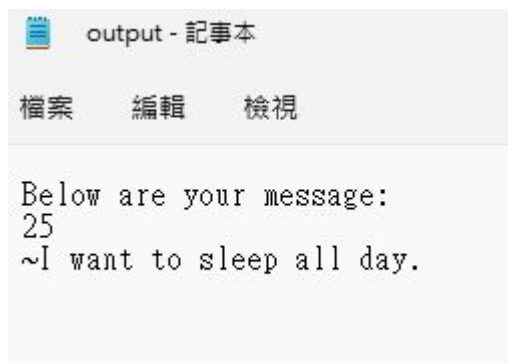
fputc(char, file pointer)

fputc() Writes a character to the file.
Ex. `fputc(c, fp);`

fputs(string, file pointer)

fputs() Writes the C string to the file.
Ex. `fputs(str, fp);`

Example



```
int main() {  
    FILE* fptr;  
    // Open the file  
    fptr = fopen("output.txt", "w");  
  
    int num1 = 25;  
  
    // Check whether the file is opened  
    if (fptr != NULL) {  
        // Write one number to the file  
        fprintf(fptr, "Below are your message:\n%d\n", num1);  
        // Write one character to the file  
        fputc('~', fptr);  
        // Write a string to the file  
        fputs("I want to sleep all day.", fptr);  
    }  
    else {  
        // If fail to open the file, print the error message  
        printf("Fail to open the 'output.txt'.");  
    }  
    // Close the file  
    fclose(fptr);  
}
```


Reposition stream position indicator

- `fseek(file pointer, offset, reference position)`
 - Sets the position indicator associated with the stream to a new position.

Constant	Example
SEEK_SET	EX : <code>fseek (fptr , 0 , SEEK_SET)</code> <code>reposition cursor(fptr, offset=0, counting from the start of the file)</code>
SEEK_CUR	EX : <code>fseek (fptr , 1 , SEEK_CUR)</code> <code>reposition cursor(fptr, offset=1, counting from the current position)</code>
SEEK_END	EX : <code>fseek (fptr , -1 , SEEK_END)</code> <code>reposition cursor(fptr, offset=-1, counting from the end of the file)</code>

Example

EOF = End Of File

main.txt - 記事本

檔案(F) 編輯(E) 格式(O)

X 123.5
Y 234.3
Z 987.1



```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<stdlib.h>

int main() {
    FILE* fptr;
    char ch;
    fptr = fopen("main.txt", "r");
    if (fptr != NULL) {
        while ((ch = getc(fptr)) != EOF) {
            ch = getc(fptr);
            printf("%c", ch);
        }
    }
}
```



Microsoft Visual

```
2.
3.
8.
```

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<stdlib.h>

int main() {
    FILE* fptr;
    char ch;
    fptr = fopen("main.txt", "r");
    if (fptr != NULL) {
        while ((ch = getc(fptr)) != EOF) {
            fseek(fptr, -1, SEEK_CUR);
            ch = getc(fptr);
            printf("%c", ch);
        }
    }
}
```



Microsoft Visual

```
X 123.5
Y 234.3
Z 987.1
```

EOF

- ◆ To check whether you have reached the end of the file (End of File, EOF)。
- ◆ In `<stdio.h>` EOF is defined as -1

Close a file



fclose()

- **fclose(fptr)**
 - Closes the file
 - Releases any memory used for the file
 - Not close the file can result in fprintf() invalid

There is a limit number of files you can open at the same time. Therefore, you need to close the unnecessary files.

Exercises

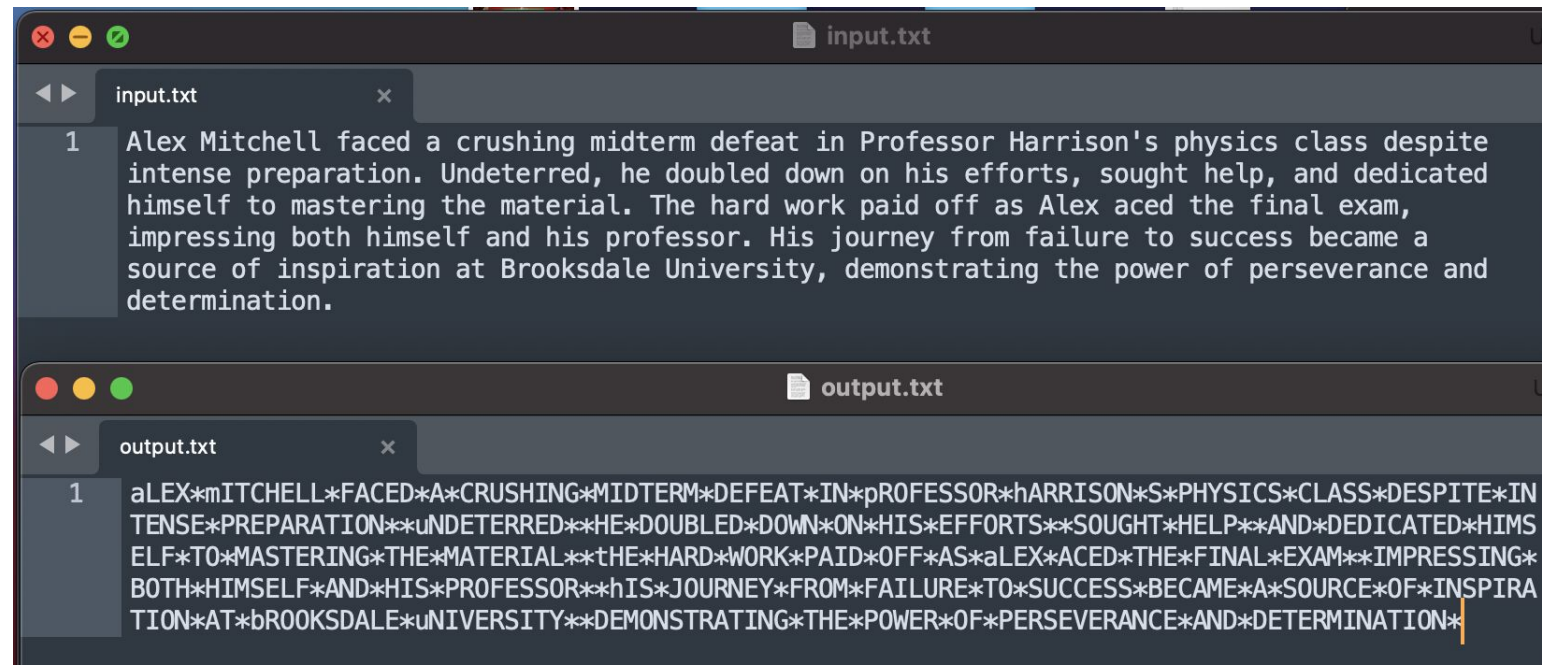


Exercise 1

- Read the “input.txt” we provide.
- Input a char C, replace non-alphabetic characters with C.
- Replace uppercase letters with lowercase and lowercase letters with uppercase.
- Create a “output.txt” to save the results.

Example

User input: *



The image shows two overlapping text editor windows. The top window, titled 'input.txt', contains a single line of text: '1 Alex Mitchell faced a crushing midterm defeat in Professor Harrison's physics class despite intense preparation. Undeterred, he doubled down on his efforts, sought help, and dedicated himself to mastering the material. The hard work paid off as Alex aced the final exam, impressing both himself and his professor. His journey from failure to success became a source of inspiration at Brooksdale University, demonstrating the power of perseverance and determination.' The bottom window, titled 'output.txt', shows the same text but with every character replaced by an asterisk (*), including the line number '1'.

```
input.txt
1 Alex Mitchell faced a crushing midterm defeat in Professor Harrison's physics class despite
intense preparation. Undeterred, he doubled down on his efforts, sought help, and dedicated
himself to mastering the material. The hard work paid off as Alex aced the final exam,
impressing both himself and his professor. His journey from failure to success became a
source of inspiration at Brooksdale University, demonstrating the power of perseverance and
determination.

output.txt
1 aLEX*mITCHELL*FACED*A*CRUSHING*MIDTERM*DEFEAT*IN*pROFESSOR*hARRISON*S*PHYSICS*CLASS*DESPITE*IN
TENSE*PREPARATION**uNDETERRED**HE*DOUBLED*DOWN*ON*HIS*EFFORTS**SOUGHT*HELP**AND*DEDICATED*HIMS
ELF*TO*MASTERING*THE*MATERIAL**tHE*HARD*WORK*PAID*OFF*AS*aLEX*ACED*THE*FINAL*EXAM**IMPRESSING*
BOTH*HIMSELF*AND*HIS*PROFESSOR**hIS*JOURNEY*FROM*FAILURE*TO*SUCCESS*BECAME*A*SOURCE*OF*INSPIRA
TION*AT*bROOKSDALE*uNIVERSITY**DEMONSTRATING*THE*POWER*OF*PERSEVERANCE*AND*DETERMINATION*
```

On OJ, you should let user input and print out the results, the input will be the following format.

```
*  
Alex Mitchell faced a crushing midterm defeat in Professor Harrison's physics class despite  
intense preparation. Undeterred, he doubled down on his efforts, sought help, and dedicated  
himself to mastering the material. The hard work paid off as Alex aced the final exam,  
impressing both himself and his professor. His journey from failure to success became a  
source of inspiration at Brooksdale University, demonstrating the power of perseverance and  
determination.
```


Hint

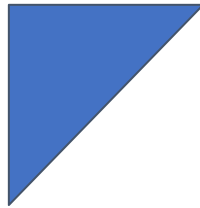
There are several ways to solve the problem.

1. Use ASCII.
2. isalpha(), isupper, islower. [reference](#)

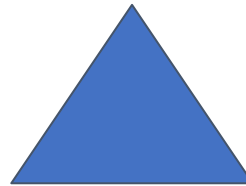
Exercise 2

- There will be an "ex2_input.txt" file, which contains two numbers in the file, representing S and C respectively.
- S represents the state to be selected, print a triangle based on different values of S in the following pattern.

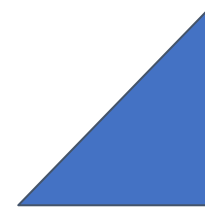
○ $S = 1$



$S = 2$

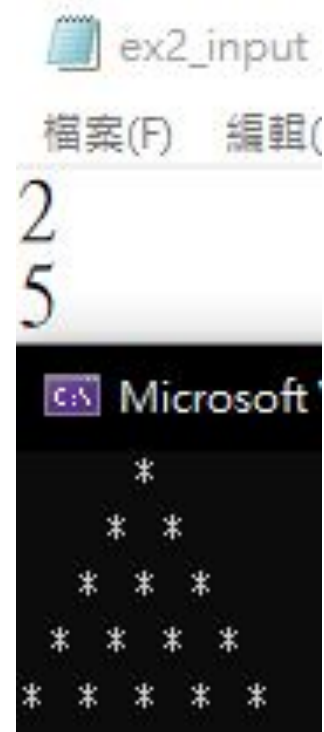


$S = 3$



- C represents the side length of the triangle ($1 \leq C \leq 10$)

There is no OJ.



Exercise Submission Format

Format:

- xxxxxxxxxx_ex_w11.zip
 - xxxxxxxxxx_ex_01.cpp
 - xxxxxxxxxx_ex_02.cpp

xxxxxxx is your student ID