

Prompts for Question I and II:

1. Upload Assignment2 file to ChatGPT and ask it “Can you solve this assignment for me?”
Then, ChatGPT asks: “Would you like help coding these solutions or any specific part of this assignment?”
2. Yes, please code the solution.

I. Text Processing

First function for counting lines, words, characters, and uppercase characters

```
#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>

void analyze_file(const char* filename) {
    FILE *file = fopen(filename, "r");
    if (file == NULL) {
        printf("Error opening file.\n");
        return;
    }

    int lines = 0, words = 0, chars = 0, upper_case = 0;
    int in_word = 0;
    char ch;

    while ((ch = fgetc(file)) != EOF) {
        // Count lines
        if (ch == '\n') {
            lines++;
        }

        // Count alphanumeric characters
        if (isalnum(ch)) {
            chars++;
        }

        // Count uppercase letters
        if (isupper(ch)) {
            upper_case++;
        }

        // Count words
        if (isspace(ch)) {
            if (in_word) {
                in_word = 0;
                words++;
            }
        } else {
            in_word = 1;
        }
    }

    // Account for EOF line and the last word
    lines++;
    if (in_word) {
        words++;
    }

    printf("Lines: %d\n", lines);
    printf("Alphanumeric characters: %d\n", chars);
    printf("Uppercase letters: %d\n", upper_case);
    printf("Words: %d\n", words);

    fclose(file);
}
```

Second function for converting characters to uppercase characters

```
void convert_to_upper(const char* input_filename, const char* output_filename) {
    FILE *input_file = fopen(input_filename, "r");
    FILE *output_file = fopen(output_filename, "w");

    if (input_file == NULL || output_file == NULL) {
        printf("Error opening files.\n");
        return;
    }

    char ch;
    while ((ch = fgetc(input_file)) != EOF) {
        fputc(toupper(ch), output_file);
    }

    fclose(input_file);
    fclose(output_file);
}

int main(int argc, char *argv[]) {
    if (argc != 2) {
        printf("Usage: %s <input_file>\n", argv[0]);
        return 1;
    }

    analyze_file(argv[1]);
    convert_to_upper(argv[1], "UPPER");

    return 0;
}
```

I. Pointers

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

int main(int argc, char *argv[]) {
    if (argc != 2) {
        printf("Usage: %s <memory_size>\n", argv[0]);
        return 1;
    }

    // Convert argv[1] to integer using atoi
    int size = atoi(argv[1]);
    if (size <= 0) {
        printf("Invalid memory size.\n");
        return 1;
    }

    // Allocate memory using malloc
    char *memory = (char*) malloc(size);
    if (memory == NULL) {
        printf("Memory allocation failed.\n");
        return 1;
    }

    // Fill the memory with 'A' to 'Z' and '1' to '9' in a loop
    char pattern[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ123456789";
    int pattern_size = 35;

    for (int i = 0; i < size; i++) {
        memory[i] = pattern[i % pattern_size];
    }

    // Print the generated string
    printf("Generated String: %.*s\n", size, memory);

    // Process memory with an int pointer and while loop
    int *int_ptr = (int*) memory;
    int bytes_processed = 0;

    while (bytes_processed + sizeof(int) <= size) {
        printf("%d : %d\n", *int_ptr, *int_ptr - 1);
        int_ptr++;
        bytes_processed += sizeof(int);
    }

    // Free allocated memory
    free(memory);

    return 0;
}
```

Prompts for generating 3 test text files:

“For Text Processing, you must create at least 3 test text files, each at least 1,000 characters, to verify the program's correctness. Can you generate these three files with a line no more than 80 characters?”

Test File 1: test1.txt

text

複製程式碼

```
The quick brown fox jumps over the lazy dog 123 times.  
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.  
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea c  
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nul  
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit  
(Repeat until the content reaches over 1,000 characters)
```

Test File 2: test2.txt

text

複製程式碼

```
1234567890 The rain in Spain stays mainly in the plain!  
Supercalifragilisticexpialidocious.  
Quis autem vel eum iure reprehenderit qui in ea voluptate velit esse quam nihil molestiae  
vel illum qui dolorem eum fugiat quo voluptas nulla pariatur?  
Praesentium, nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit,  
sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt.  
(Repeat until the content reaches over 1,000 characters)
```

Test File 3: test3.txt

text

複製程式碼

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ 0123456789. Hello, world!  
Programming is fun when you understand it.  
But as long as you keep learning, you'll always get better.  
Nullam id dolor id nibh ultricies vehicula ut id elit.  
Curabitur blandit tempus porttitor.  
Integer posuere erat a ante venenatis dapibus posuere velit aliquet.  
Aenean eu leo quam. Pellentesque ornare sem lacinia quam venenatis vestibulum.  
(Repeat until the content reaches over 1,000 characters)
```

Prompts for generating TESTPLAN:

Details is in the TESTPLAN.pdf.