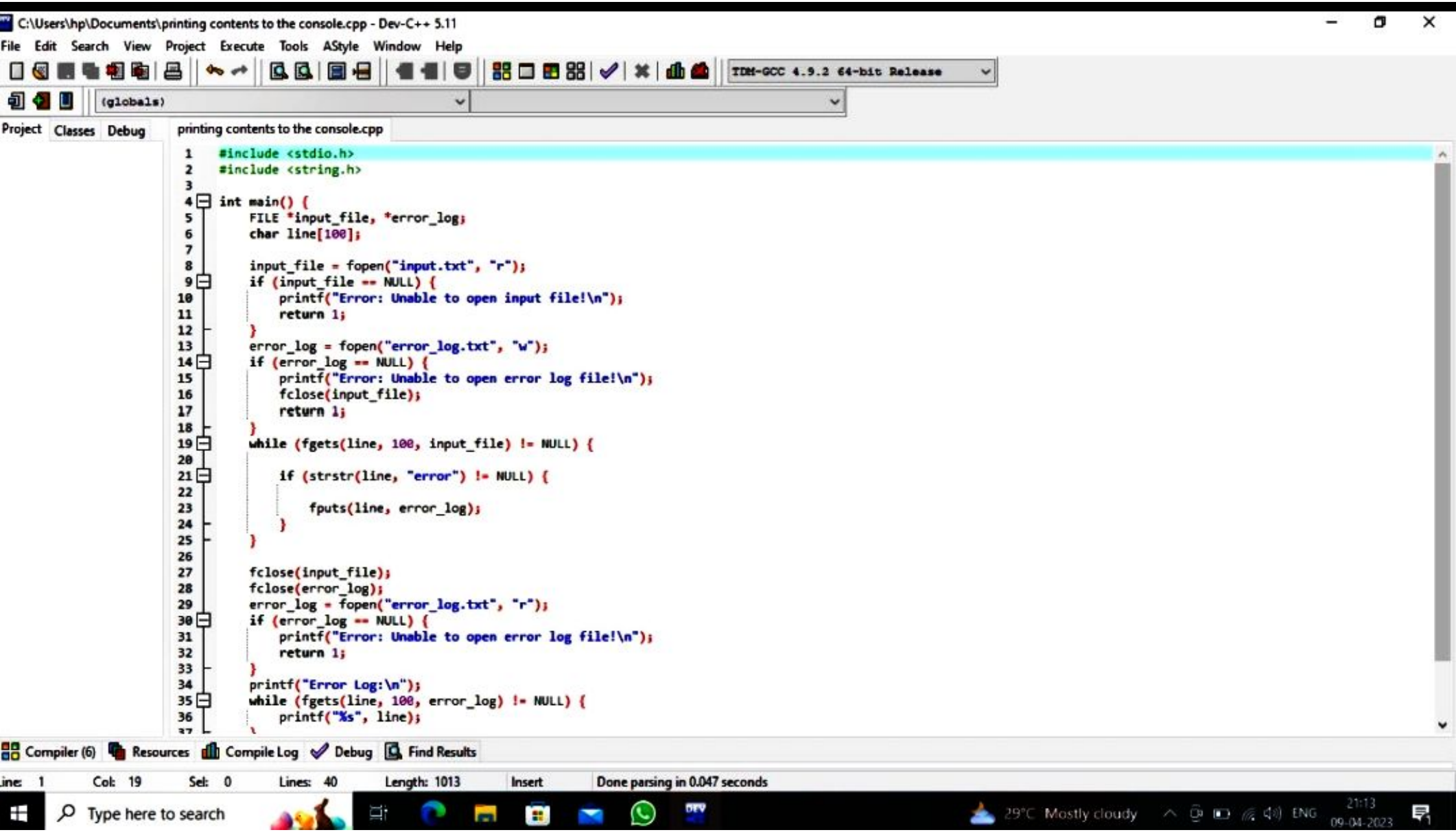
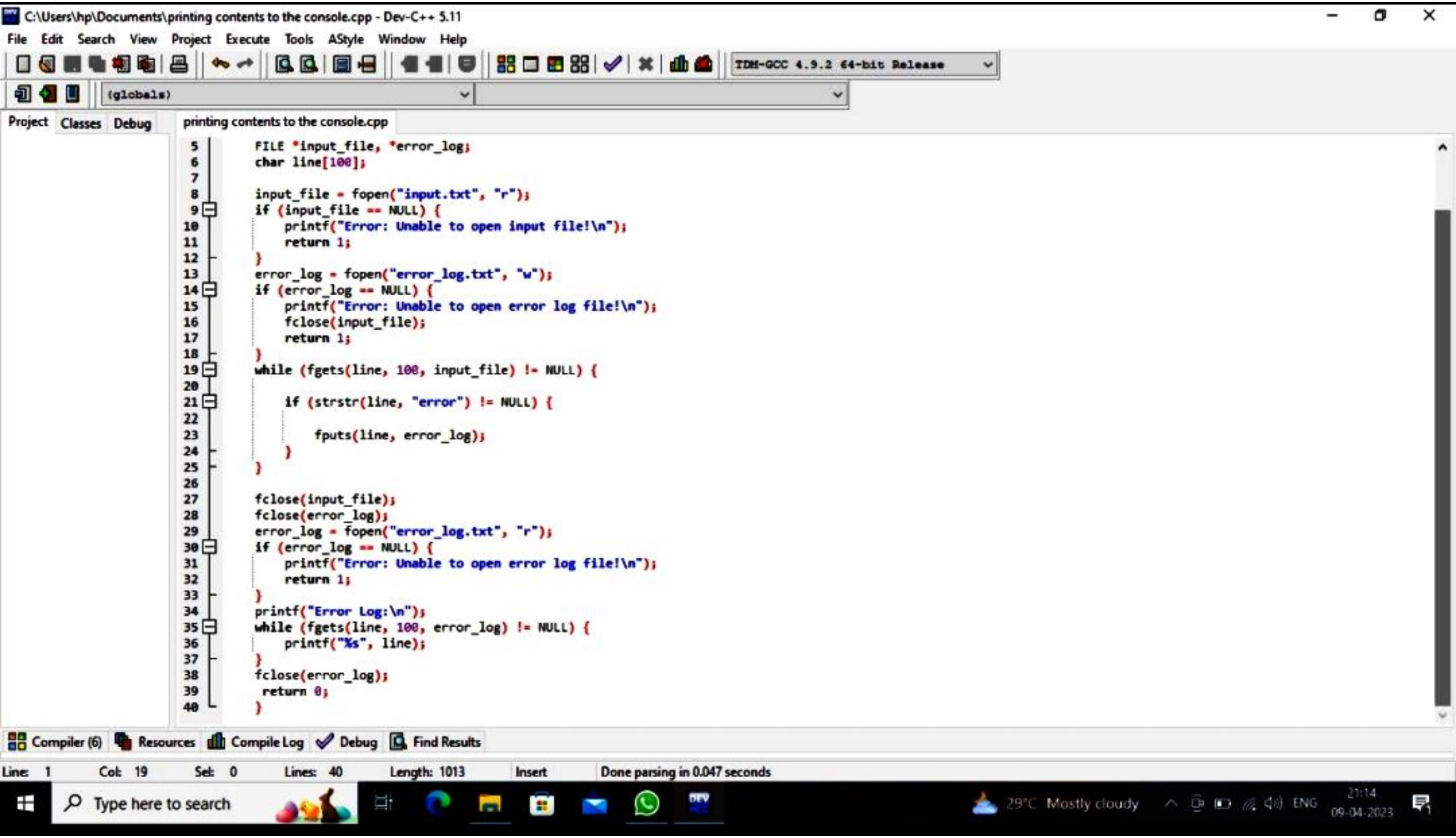
1. Write a program that reads a file named "input.txt" and writes all lines containing the word "error" to a file named "error_log.txt". Then, read the "error_log.txt" file and print the contents to the console.


C:\Users\hp\Documents\printing contents to the console.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
printing contents to the console.cpp
#include <stdio.h>
#include <string.h>
int main() {
FILE *input_file, *error_log;
char line[100];
input_file = fopen("input.txt", "r");
if (input_file == NULL) {
printf("Error: Unable to open input file!\n");
return 1;
}
error_log = fopen("error_log.txt", "w");
if (error_log == NULL) {
printf("Error: Unable to open error log file!\n");
fclose(input_file);
return 1;
}
while (fgets(line, 100, input_file) != NULL) {
if (strstr(line, "error") != NULL) {
fputs(line, error_log);
}
}
fclose(input_file);
fclose(error_log);
error_log = fopen("error_log.txt", "r");
if (error_log == NULL) {
printf("Error: Unable to open error log file!\n");
return 1;
}
printf("Error Log:\n");
while (fgets(line, 100, error_log) != NULL) {
printf("%s", line);
Compiler (6) Resources Compile Log Debug Find Results
Line: 1 Col: 19 Sel: 0 Lines: 40 Length: 1013 Insert Done parsing in 0.047 seconds
Type here to search
29°C Mostly cloudy
ENG
21:13
09-04-2023


C:\Users\hp\Documents\printing contents to the console.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
printing contents to the console.cpp
FILE *input_file, *error_log;
char line[100];
input_file = fopen("input.txt", "r");
if (input_file == NULL) {
printf("Error: Unable to open input file!\n");
return 1;
}
error_log = fopen("error_log.txt", "w");
if (error_log == NULL) {
printf("Error: Unable to open error log file!\n");
fclose(input_file);
return 1;
}
while (fgets(line, 100, input_file) != NULL) {
if (strstr(line, "error") != NULL) {
fputs(line, error_log);
}
}
fclose(input_file);
fclose(error_log);
error_log = fopen("error_log.txt", "r");
if (error_log == NULL) {
printf("Error: Unable to open error log file!\n");
return 1;
}
printf("Error Log:\n");
while (fgets(line, 100, error_log) != NULL) {
printf("%s", line);
}
fclose(error_log);
return 0;
}
Compiler (6) Resources Compile Log Debug Find Results
Line: 1 Col: 19 Sel: 0 Lines: 40 Length: 1013 Insert Done parsing in 0.047 seconds
Type here to search
29°C Mostly cloudy
ENG
21:14
09-04-2023

2. Create a program that reads data from a binary file named "data.bin" and prints it to the console in a human-readable format.

C:\Users\hp\Documents\binary file.cpp - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug

printing contents to the console.cpp [*] binary file.cpp

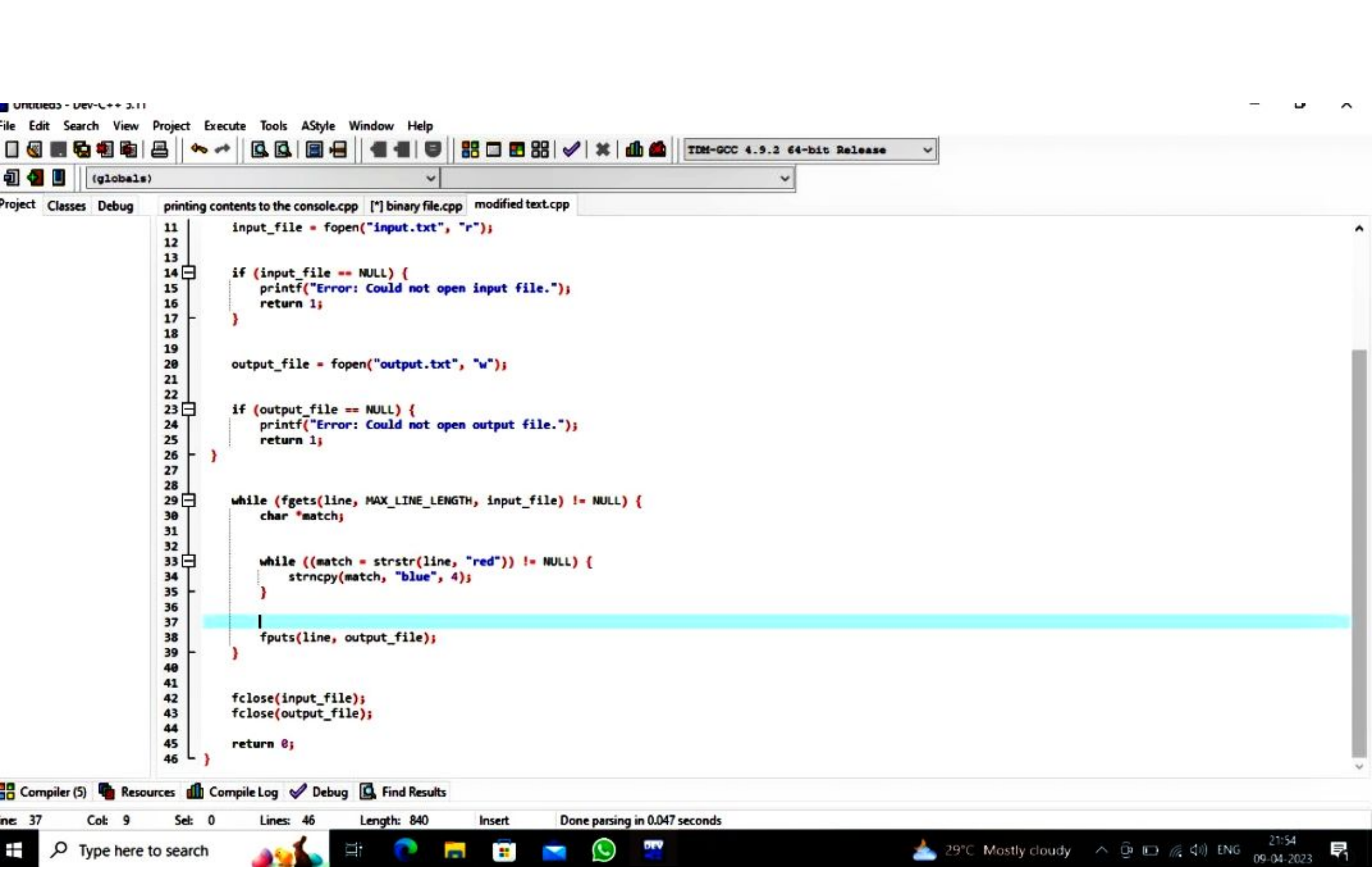
```c
#include <stdio.h>

int main()
 {
    FILE *fp;
    int num_records = 0;
    fp = fopen("data.bin", "rb");
    if (fp == NULL) {
        printf("Error: Could not open file.");
        return 1;
    }

    fread(&num_records, sizeof(int), 1, fp);
    printf("Number of records: %d\n", num_records);
    for (int i = 0; i < num_records; i++) {
        int record_id = 0;
        float record_value = 0.0f;
        fread(&record_id, sizeof(int), 1, fp);
        fread(&record_value, sizeof(float), 1, fp);
        printf("Record %d: %d, %.2f\n", i+1, record_id, record_value);
    }

    fclose(fp);

    return 0;
}
```

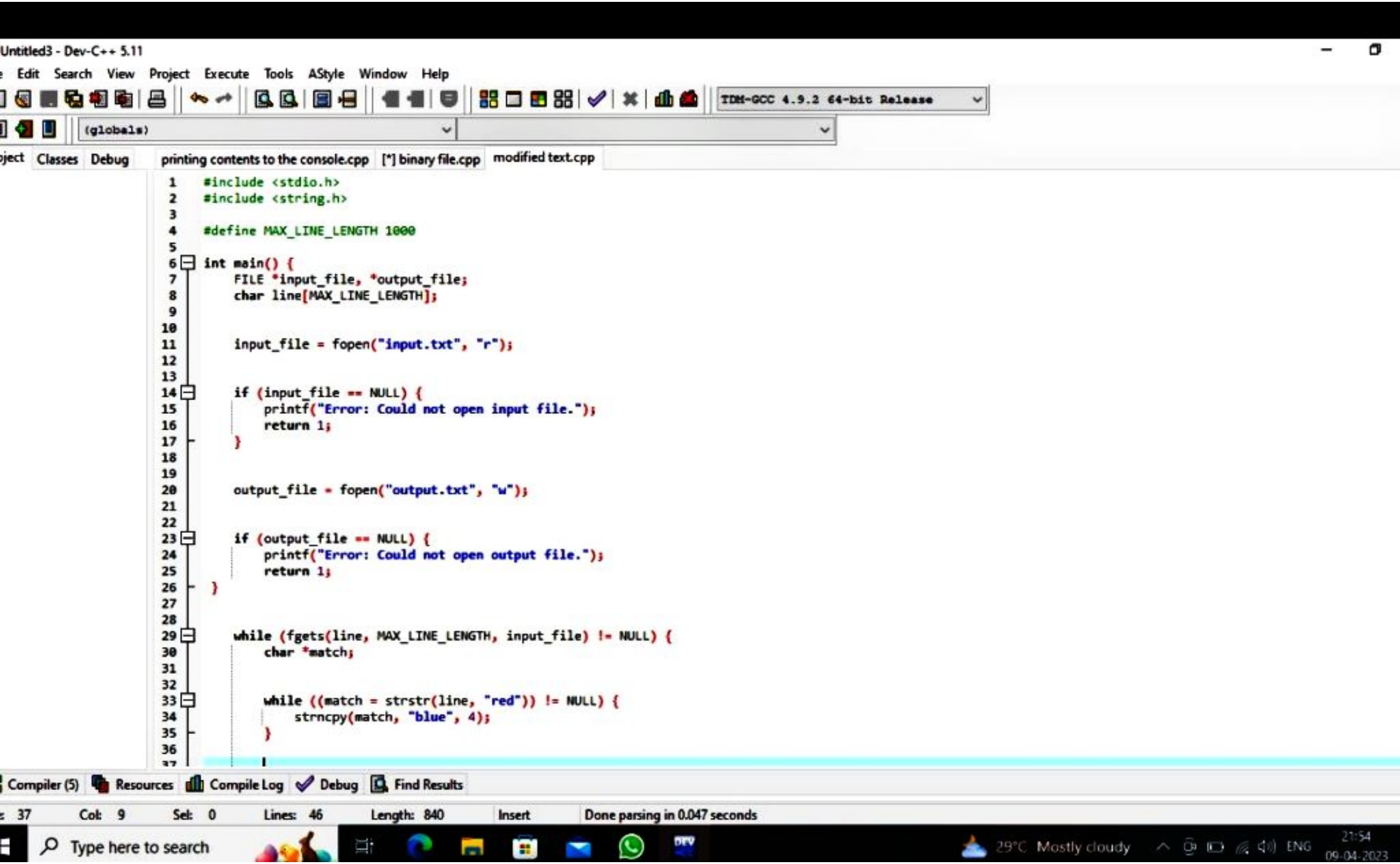
Compiler (5) Resources Compile Log Debug Find Results

Line: 1 Col: 19 Sel: 0 Lines: 26 Length: 644 Insert Done parsing in 0.031 seconds

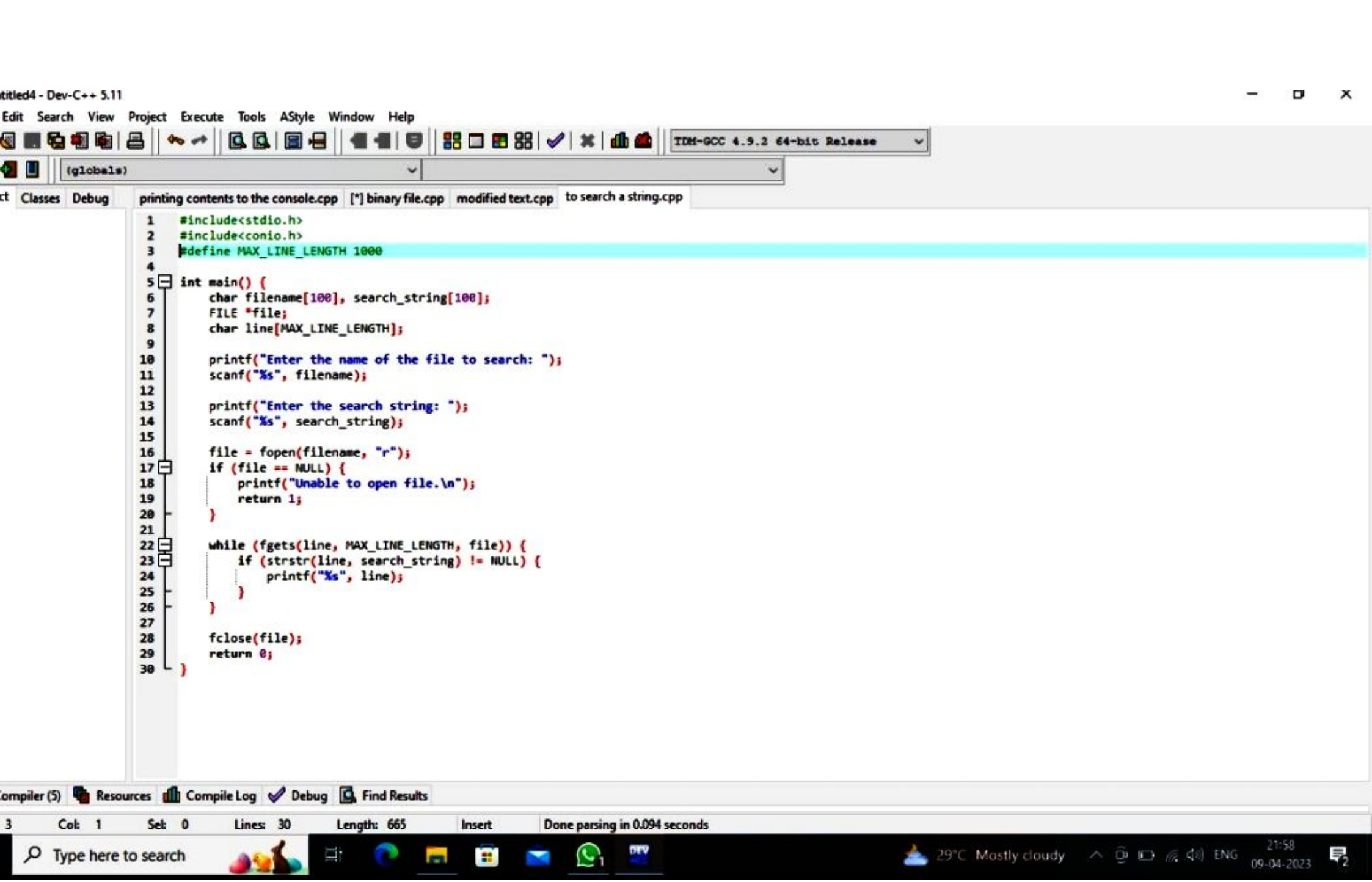
Type here to search 29°C Mostly cloudy ENG 21:30 09-04-2023

3. Write a program that reads a text file named "input.txt" and replaces all occurrences of the word "red" with the word "blue". Then, write the modified text to a new file named "output.txt".


File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Project Classes Debug
printing contents to the console.cpp
[*] binary file.cpp
modified text.cpp
input_file = fopen("input.txt", "r");
if (input_file == NULL) {
printf("Error: Could not open input file.");
return 1;
}
output_file = fopen("output.txt", "w");
if (output_file == NULL) {
printf("Error: Could not open output file.");
return 1;
}
while (fgets(line, MAX_LINE_LENGTH, input_file) != NULL) {
char *match;
while ((match = strstr(line, "red")) != NULL) {
strncpy(match, "blue", 4);
}
fputs(line, output_file);
}
fclose(input_file);
fclose(output_file);
return 0;
}
Compiler (5)
Resources
Compile Log
Debug
Find Results
Line: 37 Col: 9 Sel: 0 Lines: 46 Length: 840 Insert
Done parsing in 0.047 seconds
Type here to search
29°C Mostly cloudy
ENG
21:54
09-04-2023


Untitled3 - Dev-C++ 5.11
Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
Classes Debug
printing contents to the console.cpp
[*] binary file.cpp
modified text.cpp
#include <stdio.h>
#include <string.h>
#define MAX_LINE_LENGTH 1000
int main() {
FILE *input_file, *output_file;
char line[MAX_LINE_LENGTH];
input_file = fopen("input.txt", "r");
if (input_file == NULL) {
printf("Error: Could not open input file.");
return 1;
}
output_file = fopen("output.txt", "w");
if (output_file == NULL) {
printf("Error: Could not open output file.");
return 1;
}
while (fgets(line, MAX_LINE_LENGTH, input_file) != NULL) {
char *match;
while ((match = strstr(line, "red")) != NULL) {
strncpy(match, "blue", 4);
}
Compiler (5)
Resources
Compile Log
Debug
Find Results
37 Col: 9 Sel: 0 Lines: 46 Length: 840 Insert Done parsing in 0.047 seconds
Type here to search
29°C Mostly cloudy
ENG
21:54
09-04-2023

4. Create a program that prompts the user to enter the name of a file and a search string. The program should read the file and print all lines that contain the search string to the console.


titled4 - Dev-C++ 5.11
Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
ct Classes Debug
printing contents to the console.cpp
[*] binary file.cpp
modified text.cpp
to search a string.cpp
#include<stdio.h>
#include<conio.h>
#define MAX_LINE_LENGTH 1000
int main() {
char filename[100], search_string[100];
FILE *file;
char line[MAX_LINE_LENGTH];
printf("Enter the name of the file to search: ");
scanf("%s", filename);
printf("Enter the search string: ");
scanf("%s", search_string);
file = fopen(filename, "r");
if (file == NULL) {
printf("Unable to open file.\n");
return 1;
}
while (fgets(line, MAX_LINE_LENGTH, file)) {
if (strstr(line, search_string) != NULL) {
printf("%s", line);
}
}
fclose(file);
return 0;
}
ompiler (5)
Resources
Compile Log
Debug
Find Results
3 Col: 1 Sel: 0 Lines: 30 Length: 665 Insert Done parsing in 0.094 seconds
Type here to search
29°C Mostly cloudy
ENG
21:58
09-04-2023

5. Write a program that reads a file named "data.txt" and creates a new file named "statistics.txt". The program should write the number of occurrences of each letter of the alphabet in the file to the "statistics.txt" file.

Untitled5 - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

Project Classes Debug

printing contents to the console.cpp | [*] binary file.cpp | modified text.cpp | to search a string.cpp | reading and creating new file.cpp

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

int main() {
    FILE *fp_in, *fp_out;
    int count[26] = {0};
    char c;


    fp_in = fopen("data.txt", "r");
    if (fp_in == NULL) {
        printf("Error opening input file.\n");
        exit(1);
    }


    while ((c = fgetc(fp_in)) != EOF) {
        if (isalpha(c)) {
            count[tolower(c) - 'a']++;
        }
    }


    fclose(fp_out);

    return 0;
}
```

Compiler (5) Resources Compile Log Debug Find Results

Line: 28 Col: 2 Sel: 0 Lines: 28 Length: 467 Insert Done parsing in 0.156 seconds