GROUP PROJECT

SISTER NIVEDITA UNIVERSITY

NEWTOWN,KOLKATA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MEET THE TEAM

CHANDAN KUMAR RAJ
STUDENT

RACHAYITA DEBNATH
STUDENT

UNDER GUIDENCE OF :- PROF. RAKHI TEWARI ROY

C FILE HANDLING

WHAT IS FILE HANDLING IN C?

- In C, file handling refers to the ability to read from and write to files on a computer's hard disk. It allows a program to interact with files, either by creating new files, opening existing files, reading data from files, writing data to files, or closing files once they are no longer needed.
- File handling in C involves the use of the standard input/output library functions such as fopen(), fclose(), fprintf(), fscanf(), fread(), and fwrite(). These functions allow the programmer to manipulate files by opening them for reading or writing, reading data from them, writing data to them, and then closing them.
- The file handling functions take a file pointer as an argument. The file pointer is a data type that represents the file being manipulated. Once the file pointer is created, it can be used to perform operations on the file, such as reading or writing data to it.

- In C, file handling refers to the ability to work with files, which are collections of data stored on a computer's storage device. File handling in C involves opening, reading from, writing to, and closing files.
- The basic steps involved in file handling in C are:
  - Opening a file: In order to work with a file, you need to first open it using the fopen() function. This function takes two arguments: the name of the file and the mode in which you want to open it (read, write, or append).
  - Reading from a file: Once you have opened a file, you can read data from it using functions such as fscanf() and fgets(). These functions allow you to read data from the file in a specified format or as a string.
  - Writing to a file: If you want to write data to a file, you can use functions such as fprintf() and fputs(). These functions allow you to write data to the file in a specified format or as a string.
  - Closing a file: After you have finished working with a file, you need to close it using the fclose() function. This ensures that any changes you made to the file are saved and that the file is released from memory.
- File handling in C is essential for many programming tasks, such as working with

PROBLEM

write a program to show student name,id,marks and find the highest marks among them by using c file handling.

USING DEV C++ SOFTWARE

DATA USES IN C PROGRAMMING

SOLUTION

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

#define MAX_LEN 100
#define MAX_STUDENTS 100

struct student
{
    char name[MAX_LEN];
    char student_id[MAX_LEN];
   // char office[MAX_LEN];
    float marks;
};

int main()
{
    FILE *fp;
    char line[MAX_LEN];
    struct student students[MAX_STUDENTS];
```

SOLUTION

```
int student_count = 0;
   float max_marks = 0;
   int max_marks_student_index = 0;

   // Open the file for reading --> "r"
   fp = fopen("students.csv", "r");

   // Check if file was opened successfully
   if (fp == NULL)
   {
       printf("Failed to open file\n");
       return 1;
   }

   // Read the header line and ignore it
   fgets(line, MAX_LEN, fp);

   // Read the rest of the lines
   while (fgets(line, MAX_LEN, fp) != NULL)
   {
```

SOLUTION

```
strcpy(students[student_count].name, token);
        token = strtok(NULL, ",");
        stchar *token;
        token = strtok(line, ",");
 rcpy(students[student_count].student_id, token);
        token = strtok(NULL, ",");
       // strcpy(students[student_count].office, token);
        //token = strtok(NULL, ",");
        students[student_count].marks = atof(token);

        // Update the max marks and the corresponding student index
        if (students[student_count].marks > max_marks)
        {
            max_marks = students[student_count].marks;
            max_marks_student_index = student_count;
        }

        student_count++;
    }
```

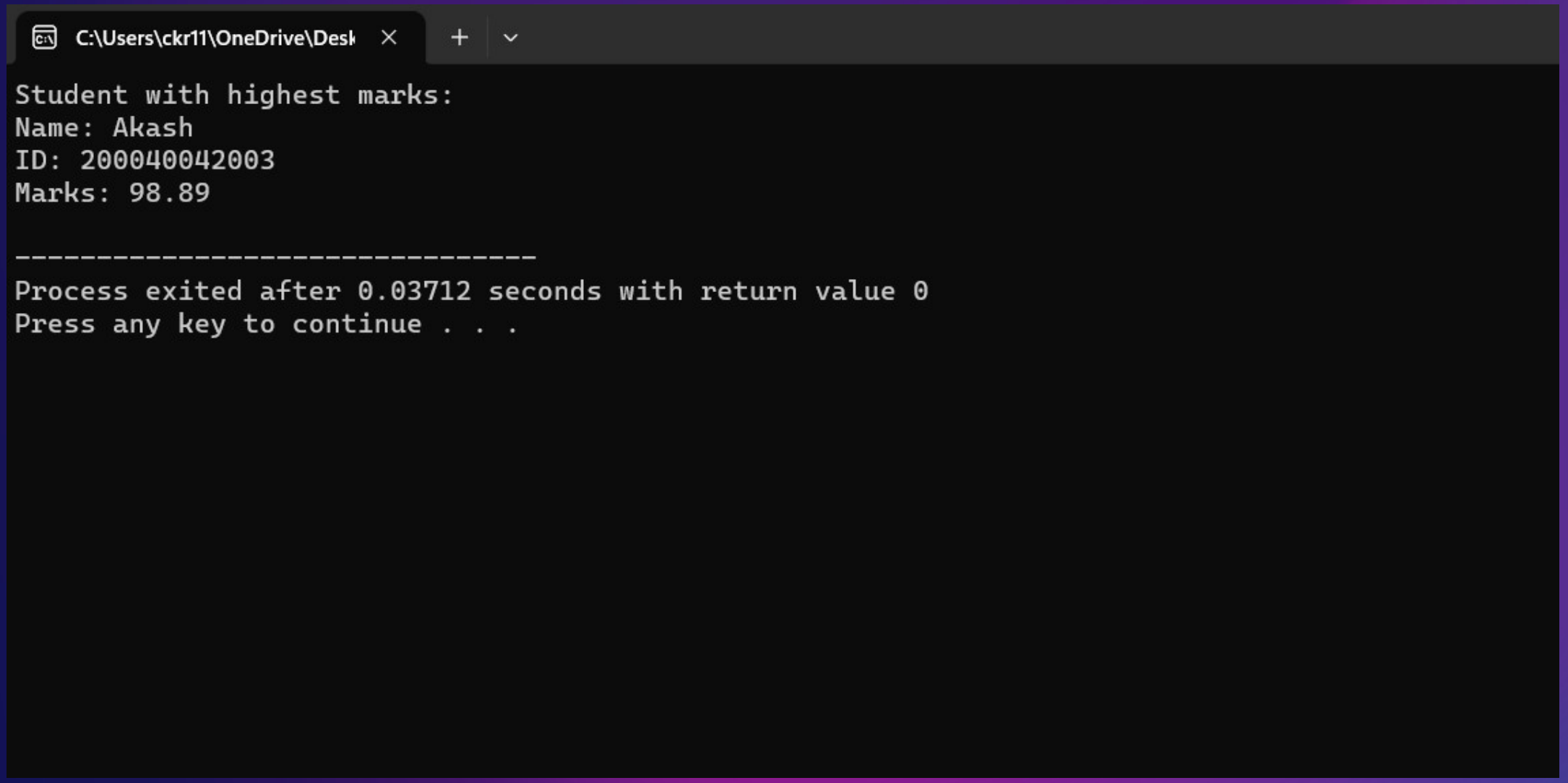
SOLUTION

```
    // Close the file
    fclose(fp);

    // Print the student with the highest marks
    printf("Student with highest marks: \n");
    printf("Name: %s\n",
students[max_marks_student_index].name);
    printf("ID: %s\n",
students[max_marks_student_index].student_id);
    //printf("Office: %s\n",
students[max_marks_student_index].office);
    printf("Marks: %.2f\n",
students[max_marks_student_index].marks);

    return 0;
```

SOLUTION


C:\Users\ckr11\OneDrive\Desk
Student with highest marks:
Name: Akash
ID: 200040042003
Marks: 98.89

Process exited after 0.03712 seconds with return value 0
Press any key to continue . . .

SPECIAL THANKS TO

MRS. SHRABANA CHAKRABARTI (HOD)

MRS. RAKHI TEWARI ROY(FACULTY)