劉　立坤

学籍番号：2IE19337P

システムソフトウェア特論演習

課題０１　説明レポート

Contents

[1. System Overview 2](#_Toc6988005)

[1.1 System Diagram 2](#_Toc6988006)

[1.2 Modules’ functions 2](#_Toc6988007)

[2. Code details 3](#_Toc6988008)

[2.1 Constants and Variables 3](#_Toc6988009)

[2.1.1 Global Constants and Variables 3](#_Toc6988010)

[2.1.2 Local Variables 3](#_Toc6988011)

[2.2 Program description 4](#_Toc6988012)

[2.2.1 express\_help\_menu 4](#_Toc6988013)

[2.2.2 struct comparing functions 5](#_Toc6988014)

[2.2.3 csv\_reader 6](#_Toc6988015)

[2.2.4 interactive\_data\_input 8](#_Toc6988016)

[2.2.5 qck\_data\_input 8](#_Toc6988017)

[2.2.6 sorting\_module 10](#_Toc6988018)

[2.2.7 statistics\_module 11](#_Toc6988019)

[2.2.8 output\_generator 15](#_Toc6988020)

[2.2.9 input\_parser 20](#_Toc6988021)

[2.2.10 Main 23](#_Toc6988022)

[2.2.11 db\_saver 23](#_Toc6988023)

[3. Operating results 24](#_Toc6988024)

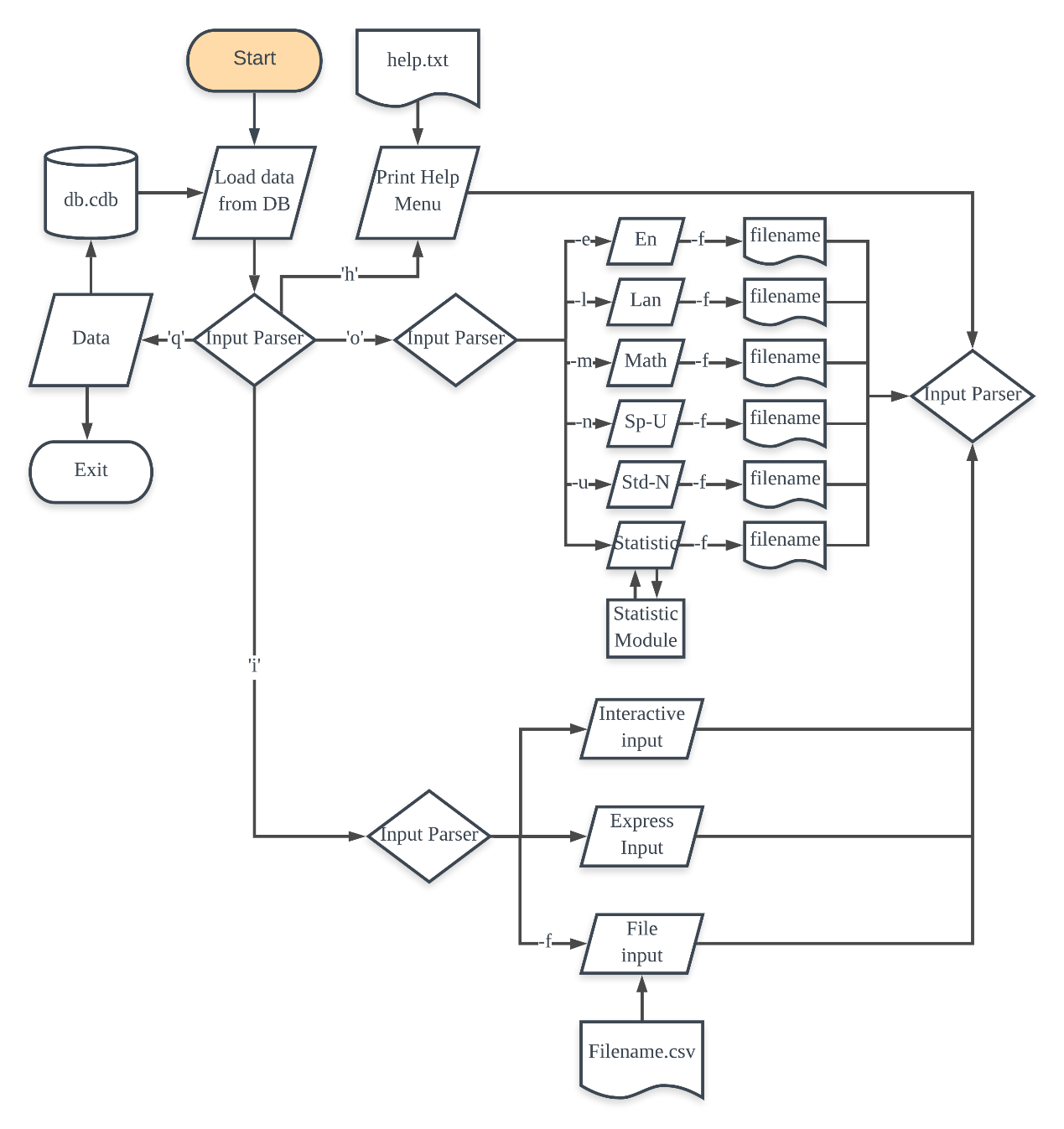
[課題01-01 24](#_Toc6988025)

[課題01-02 24](#_Toc6988026)

[課題01-03 26](#_Toc6988027)

# System Overview

## System Diagram



## Modules’ functions

|  |  |
| --- | --- |
| Module | Function |
| Input parser | Accept and analyze user’s input then pass parameters and options to each function |
| Interactive data input | Accept user’s input with prompt information |
| Express data input | Accept student information directly |
| Csv reader | Read student information from given csv file |
| Help | Print out help information |
| Statistic module | Calculating mean and standard deviation of all students, also sorting out each subject to find minimum and maximum grade |
| Output generator | Generate output according to user input, can both print to console or specified file |

# Code details

## Constants and Variables

### Global Constants and Variables

Global Constants

|  |  |  |
| --- | --- | --- |
| Constant name | Initial value | Function |
| MAX\_INPUT | 100 | Max character count for user input |
| MAX\_STD\_NUM | 50 | Max student number |
| MAX\_NAME\_LENGTH | 50 | Max length for students’ name |
| MAX\_FILE\_LENGTH | 50 | Max acceptable length for file name(input and output) |
| MAX\_FILE\_BUFFER\_SIZE | 500 | Max acceptable character count for a single line in input file |

Global Variables

|  |  |  |
| --- | --- | --- |
| Variable Name | Type | Function |
| std\_cout | int | Store the total student number |
| stats\_data | 4X4 float matrix | Store statistical data   |  |  |  |  | | --- | --- | --- | --- | | Eng\_AVG | Lan\_AVG | Math\_AVG | Total\_AVG | | Eng\_Dev | Lan\_Dev | Math\_Dev | Total\_Dev | | Eng\_MIN | Lan\_MIN | Math\_MIN | Total\_MIN | | Eng\_Max | Lan\_MAX | Math\_MAX | Total\_MAX | |
| std\_info | struct std | Storing student data:  char name: student name  int eng\_grade: English grade  int lan\_grade: language grade  int math\_grade: math grade  int total\_grade: total grade of 3 subjects |

### 2.1.2 Local Variables

|  |  |  |
| --- | --- | --- |
| Variable name | Type | Function |
| \*fp | FILE Pointer | Point to an open file for input and output |
| buf | String | File read buffer |
| std\_name | String | Storing student name |
| temp | String | Storing temporary data |
| eng\_grade | int | Temporarily storing English grade |
| lan\_grade | int | Temporarily storing language grade |
| math\_grade | int | Temporarily storing math grade |
| b\_point | int | Count the break point of loop |
| Eng\_sum | int | The total point of all students’ English grade |
| Lan\_sum | int | The total point of all students’ language grade |
| Math\_grade | int | The total point of all students’ math grade |
| En\_avg | Float | The mean of all students’ English grade |
| Lan\_avg | Float | The mean of all students’ language grade |
| Ma\_avg | Float | The mean of all students’ math grade |
| Total\_avg | Float | The mean of all students’ total grade |
| En\_stdev | Float | Standard deviation of all students’ English grade |
| Lan\_stdev | Float | Standard deviation of all students’ language grade |
| Ma\_stdev | Float | Standard deviation of all students’ math grade |
| Total\_stdev | Float | Standard deviation of all students’ total grade |
| Sq\_dev | Float | Temporary deviation during calculation process |
| True\_name\_len | Short | The actual name length of a given student |
| Input | String | Input from user |
| Usr\_input | String | User input besides command |
| Length | Short | User input’s length |
| File\_name | String | User designated file name |
| Std\_name | String | Student’s name |
| F\_flag | Short | Indicate whether file input or output command is activated. |
| Subcmd | Short | User’s input subcommand in binary format:  0001 0000 find student with specified name  0000 1000 list all students  0000 0100 list all students with English grade  0000 0010 list all students with language grade  0000 0001 list all students with math grade  000x xxxx combinational command(Not available yet, empty slot only) |

## Program description

### express\_help\_menu

Input: none

Return: none

Description:

Program’s built in help menu in case help file could not be accessed.

### struct comparing functions

Input: two struct pointers that needs to be compared

Return: integer value which indicate compare result

Description:

These functions are compare functions that required by C programming language’s qsort(void \*base, size\_t nitems, size\_t size, int (\*compar)(const void \*, const void\*)) function

### csv\_reader

Input: data source’s file name, current student in storage, data struct pointer

Return: none

Description:

This function accepts user’s designated file and read in csv data then write them into struct array and update total student count. This function is also used to read in database.



### interactive\_data\_input

Input: current student in storage, data struct pointer

Return: none

Description:

This function write prompt into console and accept user’s input one at a time.

### qck\_data\_input

Input: user’s quick input data, string’s length, current student in storage, data struct pointer

Return none

Description:

This function accepts user’s express input data and break them into name, English grade, language grade and math grade. Then it takes the extracted data and write them into struct array.



### sorting\_module

Input: current student in storage, data struct pointer, sorting option

Return: none

Description:

This function accepts current student number, data array and sorting option as parameters and then call quick sorting function and pass in required parameters.

### statistics\_module

Input: current student in storage, data struct pointer, statistic matrix pointer, file flag, file name

Return: none

Description:

This function accepts student number, student information struct array, statistic matrix pointer, file flag, file name as input.

When been called, this function first calculates the sum of each subject and calculate the average grade. The total average score is the sum of all subjects.

Description:

This part of code calculates the standard deviation of each subject as well as the deviation of total score by the equation:

then write them into the statistic matrix.

Description:

This part of code finds the minimum and maximum of each subject as well as total grade respectively by calling sort module.

It then figures out whether user is requested a file output. If requested, the code generates a csv file with specified name.

Description:

This part of code prints out the statistic result into console.

### output\_generator

Input: file flag, file name, student name, current student in storage, data struct pointer, options

Return: none

Description:

This function accepts parsed commands from user then give out desired output.

This portion of code finds out specified student and print out as file or to console according to requirements.

Description:

This option prints out all student’s name that is logged onto the system.

Description:

This part of code firstly calls for sorting module to sort all students by name to find the name with the longest length in order to determine the space that required dynamically.

It then calls for sorting module to sort students by English score on ascending order then print them out in descending order.

If the file flag is raised, this code will also write result into the specified file.

Description:

This part of code firstly calls for sorting module to sort all students by name to find the name with the longest length in order to determine the space that required dynamically.

It then calls for sorting module to sort students by language score on ascending order then print them out in descending order.

If the file flag is raised, this code will also write result into the specified file.

Description:

This part of code firstly calls for sorting module to sort all students by name to find the name with the longest length in order to determine the space that required dynamically.

It then calls for sorting module to sort students by math score on ascending order then print them out in descending order.

If the file flag is raised, this code will also write result into the specified file.

If no case was matched, the code will quit.

### input\_parser

Input: current student count, students’ information struct array, statistic data matrix pointer

Return: 1 (quit signal)

Description:

This function accepts parameters from main function and accepts external input from user then analyze user input and dispatch global variables and parameters into different function.

This particular piece of code analyzes input command:

1. “i” only: load interactive information input module.

2. “i –f XXXX”: get file name and pass parameters into csv\_reader.

3. “i” with information: analyze the express input and pass them into quick data input module.

Description:

If “q” input is detected, this part of code will return a value of one which will cause main function to exit with 0.

If “h” input is detected, this part of code will open up help file (“help.txt”) and print out information in it.

In case file isn’t exist or can’t be opened, it will load previous express help function and provide streamlined help information.

Description:

This part of code generates intended output.

If the command is “o” only, it will load statistic module with file flag = 0.

If detected file command, it will set file flag to 1 and save file name.

Then it will detect other user command and set corresponding data bits to 1 and then pass relative information to output generator.

### Main

Input: none

Return: 0 (when quit normally)

Description:

The main function is the program’s entrence. It initializes global student count, statistic data matrix and students’ information struct arry.

When executed, it will firstly print out initial information then calls input\_parser and check for its return value with a dead loop. If detect a “1” as return, the program will end and return with code “0”.

### db\_saver

Input: none

Return: 9990 (Database written error)

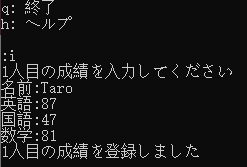
Description:

This function save current data in student struct into db file

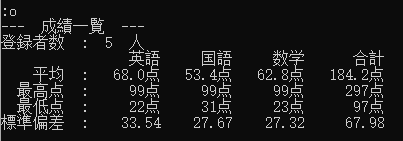
# Operating results

## 課題01-01

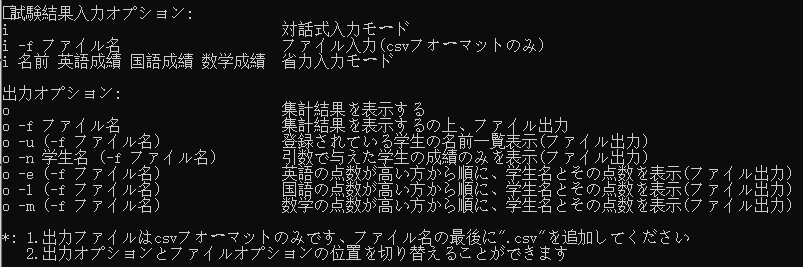
Grades input via dialog



Statistic result



Help menu



quit

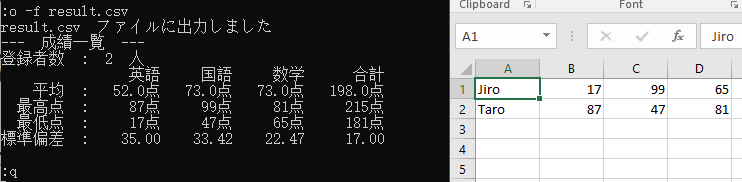


## 課題01-02

Express input function



Statistic output with file saving

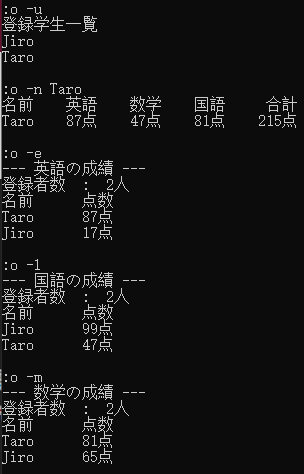


File input function



## 課題01-03

5 output option



5 output option with file output function

