Hong Kong Certificate of Education Examination 2009

Computer and Information Technology Paper 3 (Coursework)

Module A: Algorithm and Programming

Title: Multiple-Choice Analysis Report

Candidate ID No: X0000000 (5D11)

Contents

Chapter 1 Introduction

1.1 Background

There is an interschool mathematics competition held recently, which is answered in the form of multiple choice question. The answer of the competition is then collected by answer sheets, and converted into text (.txt) by an Optical Mark Recognition (OMR) system.

I am a programmer which developing a program which can read the data converted by OMR system, and mark all the raw data from OMR system, the final data will be produced into an analysis report.

1.2 Aims

The main feature that I want is convenience!

The program simply aims to produce a meaningful and readable competition report for the analytical use for school and mathematics organization, and to include in the mathematics competition newsletter.

To make the program meaningful, the followings items will be included in the report:

1. General items
    - 1.1. Total numbers of the participants
    - 1.2. Total numbers of participating schools
    - 1.3. Total numbers of participants in each participating school
2. Prize
    - 2.1. Winners of individual awards

2.2. Winners of school awards

3. Analysis

3.1. Analysis of each question

3.2. Percentage of each choice of answer

1.3 Program Development Plan

Before the start of writing a program, a MC text file must be prepared, therefore, we may either use a MC answer generator to emulate the real situation for competition, or to type the MC answer by myself which could not avoid format and typing error.

For simplicity, I choose to use MC generator which is an Excel script.

After I have decided to use a MC generator, I will modify the formula into optimum format for my program .I found that Excel file is unfavorable for me to run in Pascal, therefore I copied the MC answer into notepad and covert into text file for my convenience.

However, participants' info should be input in another text file manually.

As the structure of the input files could be complicated, it will be discussed in the later chapter.

The next step is to define the criteria of getting awards, and the details are listed in chapter 2.1.

In terms of the structure of the program, it will consist of four main parts:

1. Input of data

This part will consist of all the data which should be process, some of them will compile into files and input into the program, some of them will be inputted manually via keyboard.

2. Processing of data

This is a very crucial stage in the program which use up a lot of programming code, and is the core of this program.

It will consist of logical comparison of the MC answer and answer key, summing up final score for participants, sorting of participants according to their score, identify the winners and calculate the correct percentage of each MC question.

It will be complicated, so it will be discussed in a separated chapter.

3. Output of data

It is the final part of the program, and the only part which is useful to the user. This part contains analysis of MC questions, participants' score, school name and winners' awards.

The output data will both show on the screen and can be saving into a single text file which contains all the details the user should know.

The structure of output file format will be discussed further in chapter 3.4.

4. Functions and algorithms

This part is actually included in processing of data, but I would like to mention it specifically.

The main functions of the program are to mark the answer for participants and analyze it.

It is known that the first part of the program is to collect the data, after the collection of data, it required a comparison between answer from participants and the answer key, and then the program can record all the score of the participants into variable.

After that, the program will sort the participants into order according to their marks via "Bubble Sort", and then the program can easily identify the participants with highest marks and assign awards to them.

After all participants had included, the program will summing up and calculate the correct percentage of each question, this provide convenience to the user.

All of the result will record into certain variable and put into the output report at last.

After the construction of program, the program will undergo a testing and debugging process which could remove most error in the program. The general procedure of debugging process will be testing using random file.

Also, most malfunction or weakness in the program will be corrected and improved.

1.4 Summary

In this programming project, I learned to use Pascal and understand its advantages and disadvantage, and some general features of other programming language.

However, I do understand Pascal contain many weakness which make it less powerful than other programming language. I will still continue to make good use of this programming language.

In terms of further improvement of the program, I would like to try to use a new type of programming language which could be harder and more useful. I will also include some new feature in this program, such as graph, bar chart for analysis, etc.

Chapter 2 Analysis

2.1 Competition Regulations

In order to simplify the regulations of this competition, the regulation will be listed in point form.

The mathematics competition has the followings regulation and restriction:

1. All participants **should** participate on the behalf of his or her school.

2. No more than **10** schools can participate at a time.

3. No more than **10** participants can participate on behalf of **one** school.

4. There are **two** sets of MC questions, A & B.

5. All questions **should** be answered.

6. There are **5** choices in the MC questions, **A, B, C, D** and **E**.

7. All participant should be **form 3** or above.

8. Every question carries **1** mark.

9. This is the **final** round of the mathematics competition, all result will be counted.

10. There will be **four** kinds of individual awards.

    10.1 Gold medal for the one score highest marks.

    10.2 Silver medal for the one score the second high marks.

    10.3 Bronze medal for the one score the third high marks.

11. There will be one kind of school awards for the school with highest overall marks.

2.2 Data Collection

1. All answer should be put on the OMR MC Answer sheet.

2. All answer sheet will be read be OMR system and then converted into text file.

3. As there should be no more than 100 participants, at most 100 MC answer sheet will be collected.

4. All result from the participant will be compiling into a single file.

2.3 Input, Output and Process

1. The input data will consist of

    1.1 School names and school codes

    1.2 Participants code

1.3 Participants’ answer

1.4 Question ser code

1.5 MC answer key.

2. All data will be included in **two** text (.txt) file.

1.6 A file for storing answer key.

1.7 And a file for storing other information such as MC answers.

3. Data will be inputted in to data file by keyboard, except MC answer.

4. Data will then compile with MC answer and input into the program.

5. Output report will be a text (.txt) file and on-screen output.

6. The output report will consist of

1.8 Total numbers of participants.

1.9 Total numbers of participating school

1.10 Numbers of participant in each participating school.

1.11 Participant code.

1.12 School name.

1.13 School code.

1.14 Awards to winning unit.

1.15 Passing rate.

1.16 Score percentages for each question.

1.17 Highest score.

7. The output file will show on the same directory as the program with a fixed file name.

8. Main process in the program will include

   1.18 Calculation

   1.19 Function

   1.20 Logical operation

   1.21 Sorting

   1.22 Reading and writing text file

2.4 Choice of IT Tools

This program is developing to operate under a Dos-based operating system, eg. Windows and Dos, therefore the developer should be focus on develop a Dos-based program.

There are many kinds of programming language among the operating system, therefore the program can have a variety of choice.

Here is some of the example of programming language:

1. Pascal

2. Basic

3. Visual Basic

4. C/C++

5. Java

The reason that Pascal is chosen to be the programming language is complicated, mainly due to its simplicity and it original educational purpose.

Further information and comparison please refer to following table:

Advantage & Disadvantage of different programming language

Programming language	Advantage	Disadvantage
Pascal	-User friendly -Easy to learn	-Limited application -No multi-platform support
Basic	-More function available than Pascal -Relatively easy to learn	-No GUI support
Visual Basic	-Provide GUI support -Easier to learn than Basic	-Lower capability in non-graphical OS
C/C++	-Powerful function -Multi-application	-Complicated -Hard for beginner programmer -No Boolean type in C
Java	-Powerful function -Multi-platform favorable	-Complicated

In conclusion, I choose to use Pascal because it is simple and easy to learn, also its resources are readily available, therefore a perfect programming language for creating this program.

Moreover, I have to choose a Pascal compiler for me to compile my program into a executable exe file.

Here is some of the example of famous Pascal compiler that I can choose from:

1. Quick Pascal

2. Turbo Pascal

3. Dev-Pascal

Both three compiler that I know possess different feature, and there feature are shown in the following table.

Program compiler	Feature
Quick Pascal	A Dos-based compiler which is not so user-friendly, but still capable to develop a program.
Turbo Pascal	A Dos-based compiler which run faster than Quick Pascal, and with better support of development.
Dev-Pascal	A GUI-based compiler which cannot run in pure Dos environment, but it is much more user friendly than the two compiler shown above.

To conclude, Dev-Pascal is the best for my program development, as it is more user-friendly, therefore I choose to use Dev-Pascal for my programming project.

To provide a programming development environment, I choose to use a standard personal computer with Windows XP installed, as this operating system is capable to run Dos-based program with cmd.exe or MS-DOS.

For preparing data to input into the program, I will need the aid of keyboard and mouse. In terms of software, I will need the aid of Notepad and WordPad.

2.5 Conclusion of Study

I have studied many of the programming language, which I can finally find a suitable programming language –Pascal for my program development. I have spotted out the feature of this programming language, both the advantages and disadvantages.

In the later time, I have also found a suitable compiler –Dev-Pascal to compile my program into executable exe file. This program compiler is the best Pascal compiler among the compiler that I know.

Chapter 3 Design of Solution

3.1 Brief Description

In this chapter, I will give a brief description on what I am going to do in order to solve the problem.

First of all, the problem is to mark the MC answer from the participants of the mathematics competition. Therefore the general procedure will be as follows:

1. Collect all the MC answer sheets from the participants after the mathematics competition.

2. Precede the MC answer sheets to the OMR system.

3. The OMR system will then generate a single text file.

4. The text file, information file and MC answer key will be input into the program.

5. The program will first calculate the marks of all individual participants then schools.

6. The program will then sort all the result according to their result.

7. After that, the program will find all individual winners, merit awards and school award.

8. Finally, the result will show in the program and can be saving into a text file report which includes all participants, school result, awards and question analysis.

3.2 Refinement of Problem

There are several problems I may encounter during constructing this program; therefore I may want to figure out these problems before I have finished this program.

I will state the problems that I may encounter as follows:

As the MC analyzer consists of many parts, Input, Process and Output, it require us to separate these task and solve it individually.

The input part is about data that the user has to input to the program, however some data are generated by OMR, therefore the data that the user has to input is the file name of the give text file.

The text that the user need to input is anskey.txt, stu_ans.txt and sch_info.txt, this include all data which will be process by the program.

The process part can be further split into many parts.

The first part is to assign and identify the data in the files, after this step, the data of school information, student information and answer.

This process step also include matching of information, for example, the school code in student file will match the school name in the school file to output the data, and make it more readable.

The program will also compare the answer of student and the model answer, then calculate how many marks student and school will get.

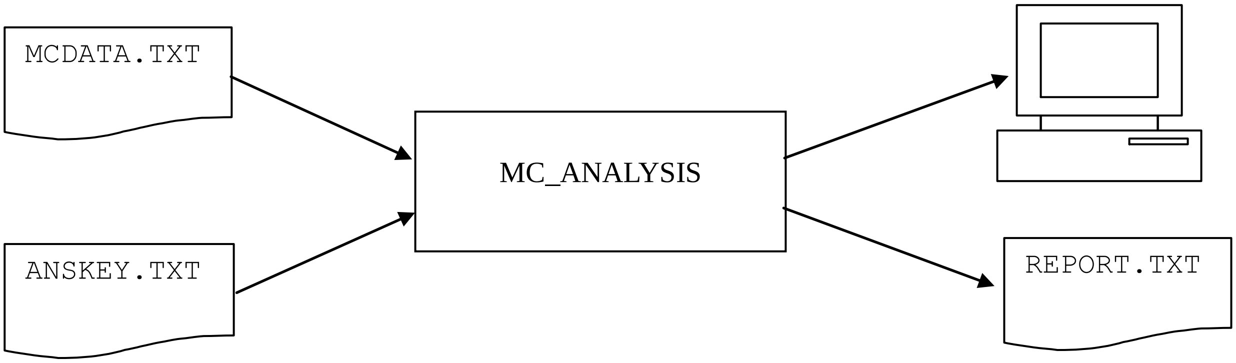
The marks value obtain in the previous stage will be used for sorting of student, hence find the winner

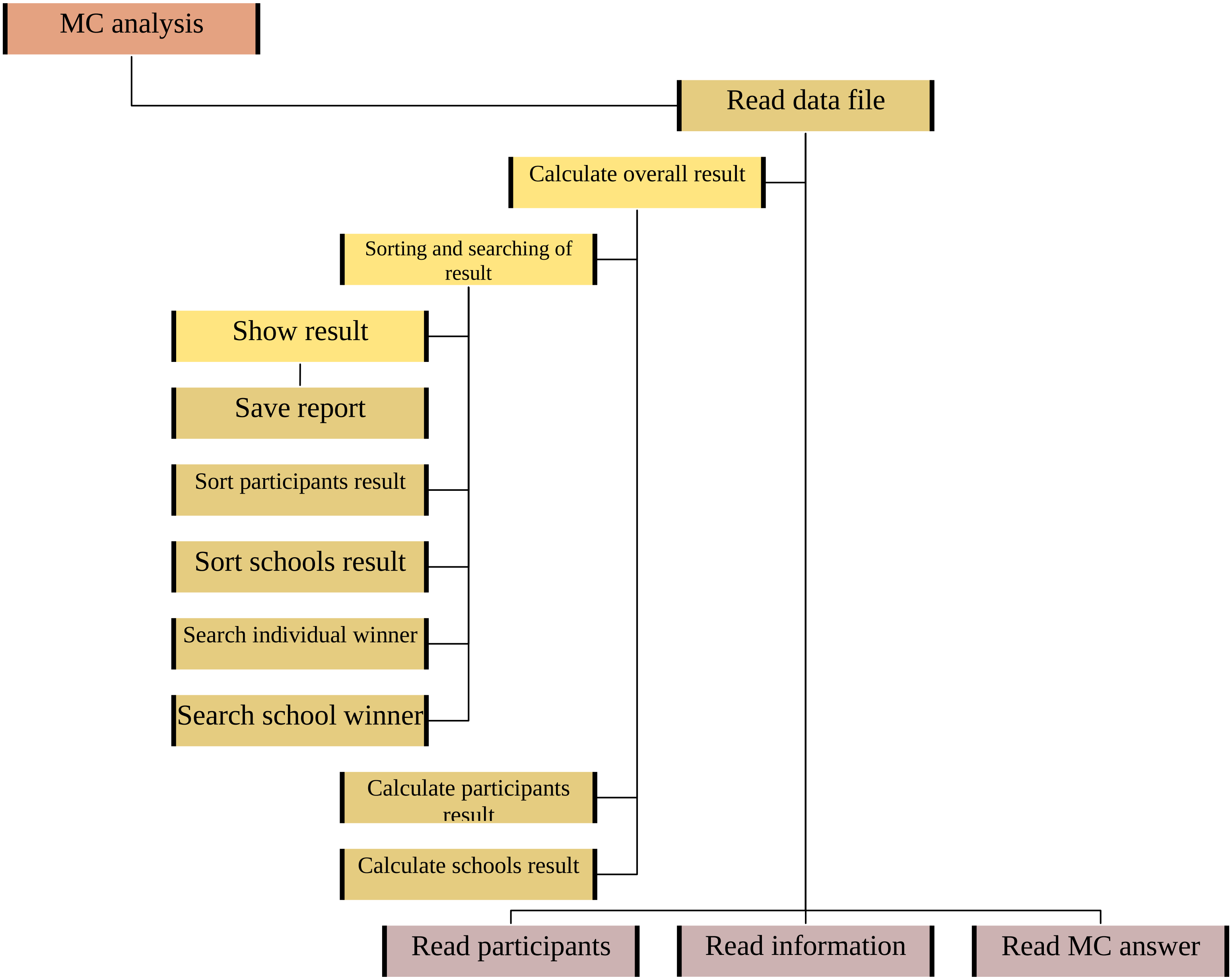
The average marks and student marks obtained will also used to plot chart, and to give awards.

In the last part, the output part is about how the data is shown to the user, the program provide two form of data output, On-screen output and text file output.

The On-screen output usually is used for a quick reference for user, and the text file is used for further analysis.

The output is an easy task, as all data has been collected and processed, so the the data can output onto screen or into text file.





3.3 Input Data File Formats

As mentioned before, there are three file needed to input to the program, sch_info.txt, stu_ans.txt and anskey.txt.

The first file, sch_info.txt, contain school code and school name, the format will present in following table.

e.g.

School Code (3 characters)	School name (8 characters)
001	a_school
002	b_school
003	c_school
004	d_school

The second file is stu_ans.txt, this file contain school code, student code, student name and MC answer of students, the format will present in the following table.

e.g.

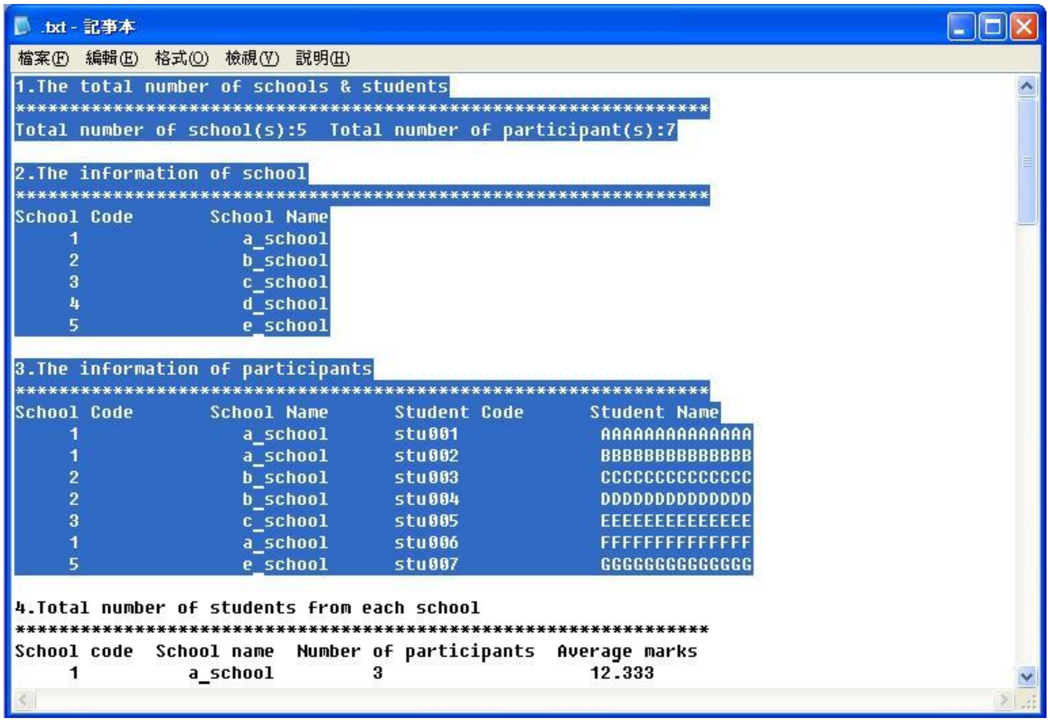
School Code (6 characters)	Participant Code (6 characters)	Participants Name (15 characters)	Answers of 50 MC questions 50 characters in ‘A’, ‘B’, C’, ‘D’, ‘E’								
001	stu001	AAAAAAAAAAAAAAA	A	C	A	C		E	D	A	E
002	stu002	BBBBBBBBBBBBBBB	A	A	B	B		D	E	C	A
003	stu003	CCCCCCCCCCCCCCC	B	E	C	D		D	B	C	A
004	stu004	DDDDDDDDDDDDDDD	D	E	B	B		A	C	A	A

The final file that needed to be inputted is anskey.txt, it consist only the answer key and nothing. The format is simple and shown below.

Answers of 50 MC questions 50 characters in ‘A’, ‘B’, C’, ‘D’, ‘E’								
A	C	A	C		E	D	A	E

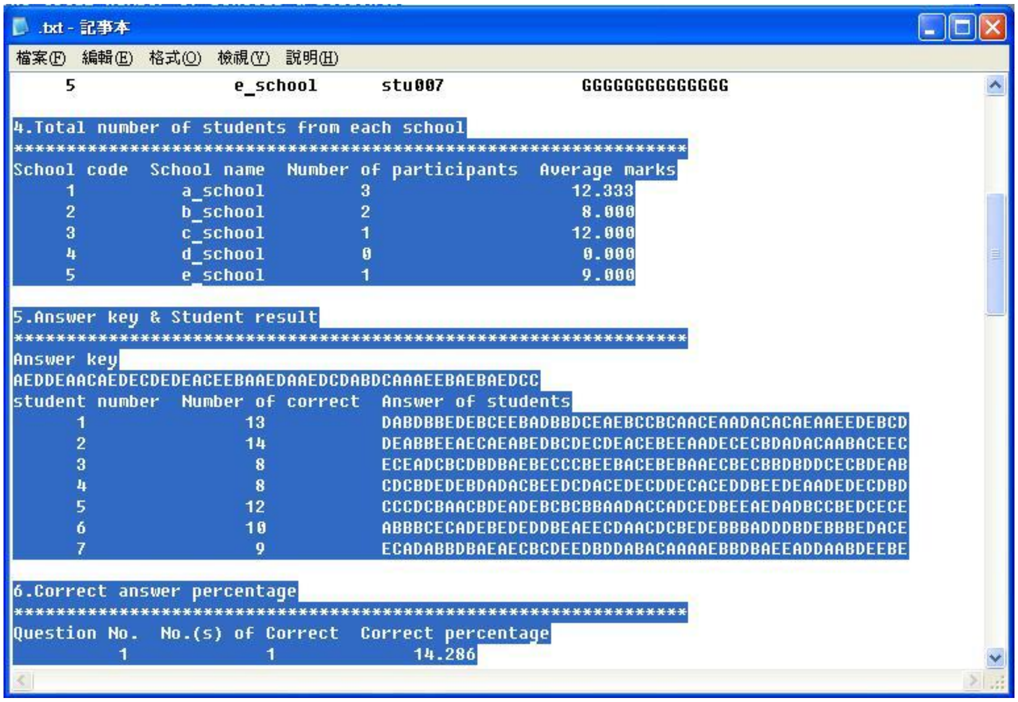
3.4 Output Report Format

The output report consists of many items that is not easily to describe, so I will describe with the aid of pictures.



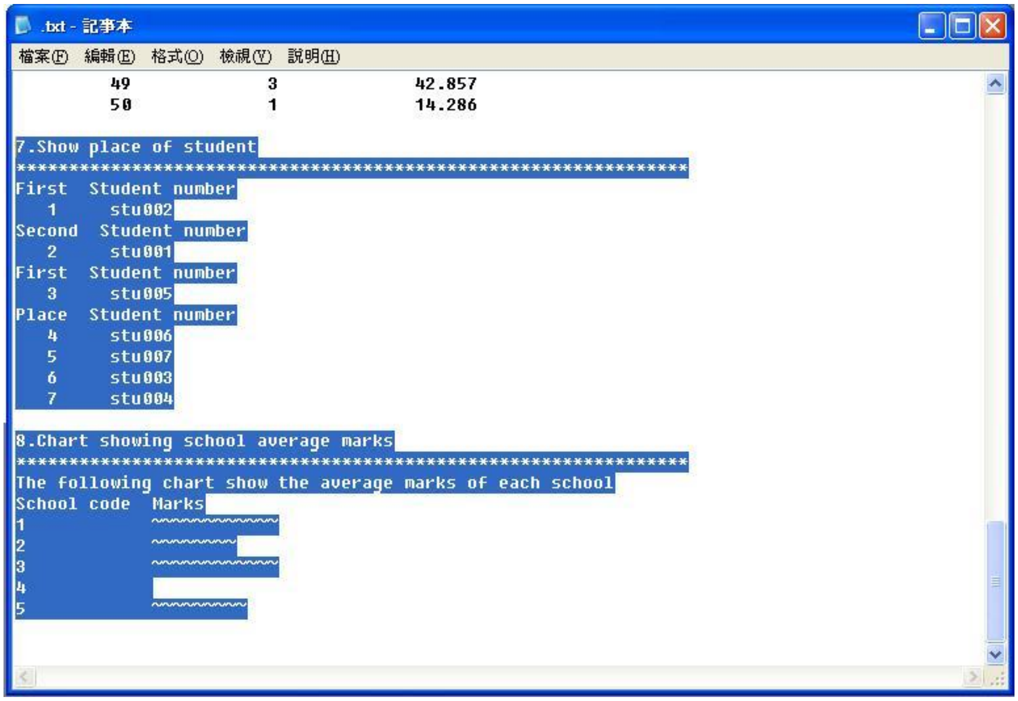
```
1.The total number of schools & students
****************************************************************
Total number of school(s):5  Total number of participant(s):7

2.The information of school
****************************************************************
School Code       School Name
     1               a_school
     2               b_school
     3               c_school
     4               d_school
     5               e_school

3.The information of participants
****************************************************************
School Code       School Name      Student Code      Student Name
     1               a_school      stu001             AAAAAAAAAAAAAA
     1               a_school      stu002             BBBBBBBBBBBBBB
     2               b_school      stu003             CCCCCCCCCCCCCC
     2               b_school      stu004             DDDDDDDDDDDDDD
     3               c_school      stu005             EEEEEEEEEEEEEE
     1               a_school      stu006             FFFFFFFFFFFFFF
     5               e_school      stu007             GGGGGGGGGGGGGG

4.Total number of students from each school
****************************************************************
School code  School name  Number of participants  Average marks
     1          a_school         3                    12.333
```

In this picture, 3 items were highlighted, these were the data I have inputted in separate 3 files, the program then analyze and combine the data into one file.



```
     5               e_school      stu007            GGGGGGGGGGGGGG

4.Total number of students from each school
****************************************************************
School code  School name  Number of participants  Average marks
     1          a_school          3                    12.333
     2          b_school          2                     8.000
     3          c_school          1                    12.000
     4          d_school          0                     0.000
     5          e_school          1                     9.000

5.Answer key & Student result
****************************************************************
Answer key
AEDDEAACAEDECDEDEACEEBAAEDAAEDCDABDCAAAEEBAEBAEDCC
student number  Number of correct  Answer of students
      1               13           DABDBBEDEBCEEBADBBDCEAEBCCBCAACEAADACACAEAAEEDEBCD
      2               14           DEABBEEAECAEABEDBCDECDEACEBEEAADECECBDADACAABACEEC
      3                8           ECEADCBCDBDBAEBECCCBEEBACEBEBAAECBECBBDBDDCECBDEAB
      4                8           CDCBDEDEBDADACBEEDCDACEDECDDECACEDDBEEDEAADEDECDBD
      5               12           CCCDCBAACBDEADEBCBCBBAADACCADCEDBEEAEDADBCCBEDCECE
      6               10           ABBBCECADEBEDEDDBEAEECDAACDCBEDEBBBADDDBDEBBBEDACE
      7                9           ECADABBDBAEAECBCDEEDBDDABACAAAAEBBDBAEEADDAABDEEBE

6.Correct answer percentage
****************************************************************
Question No.  No.(s) of Correct  Correct percentage
          1             1             14.286
```

In the second picture, 3 more items were highlighted, these item are marks and percentage, etc. These information should be analyzes by the program as it does not include in the input file.

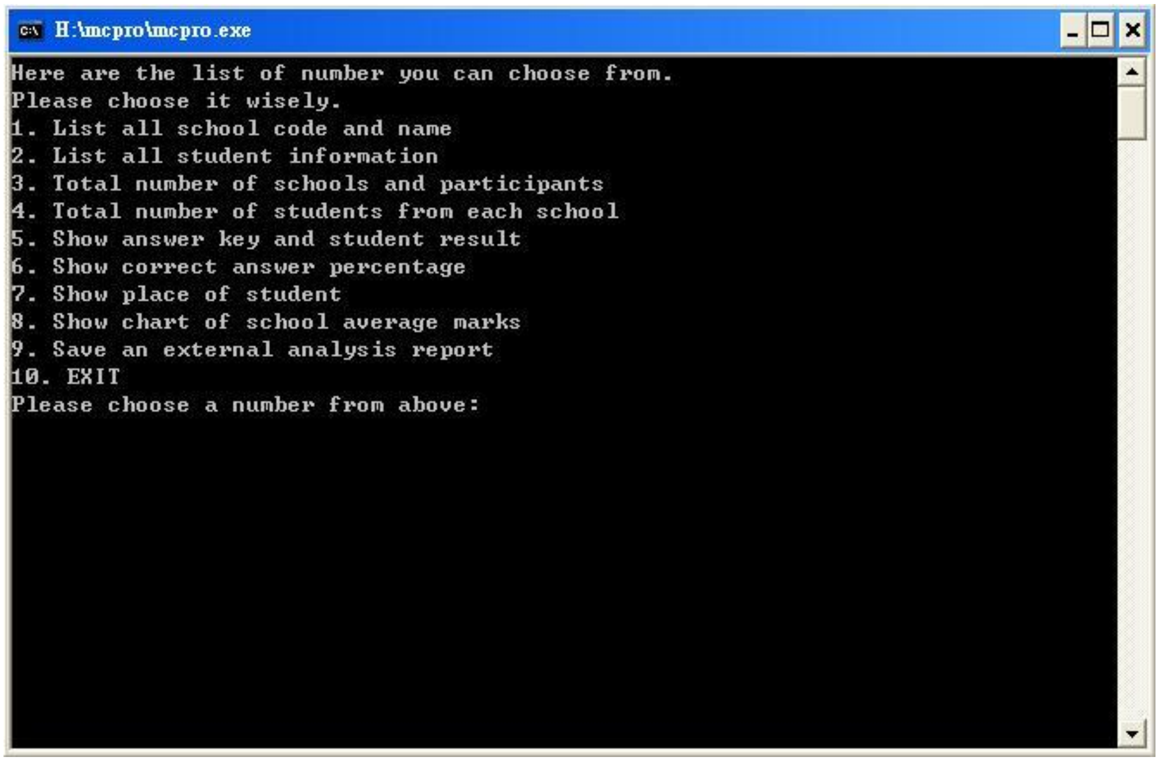


```
         49             3             42.857
         50             1             14.286

7.Show place of student
****************************************************************
First  Student number
   1      stu002
Second  Student number
   2      stu001
First  Student number
   3      stu005
Place  Student number
   4      stu006
   5      stu007
   6      stu003
   7      stu004

8.Chart showing school average marks
****************************************************************
The following chart show the average marks of each school
School code  Marks
1            ~~~~~~~~~~~~
2            ~~~~~~~~
3            ~~~~~~~~~~~~
4
5            ~~~~~~~~~
```

The last 2 items are about the place and winners of student, also a chart of school average marks are provided to give a clear result.

In the other hand, analysis report can also be shown on On-screen, which means in the program. However, the limitation of the function is that it is temporary result and it is not well-organized as it is limited by screen size.

The following picture show the menu of the program, the output format is generally as same as the output text file.



```
Here are the list of number you can choose from.
Please choose it wisely.
1. List all school code and name
2. List all student information
3. Total number of schools and participants
4. Total number of students from each school
5. Show answer key and student result
6. Show correct answer percentage
7. Show place of student
8. Show chart of school average marks
9. Save an external analysis report
10. EXIT
Please choose a number from above:
```

Chapter 4 Implementation

4.1 Brief Description

In this chapter, the actual construction of the program will be discussed.

Before constructing the program, what we need is to define the constant and variable that will used by the program. The variable could be global or local. Some of the variables are array as it is used to contain same kind of information but from different object. Array provides convenience for storing information.

Here is the list of global variable:

```
Const

     max_part=100;       {maximum number of participants}

     max_sch=10;          {maximum number of schools}

     max_question=50; {maximum number of questions}

Var

   sch_code:array [1..max_sch] of integer;                {school code}

   sch_code_stu:array [1..max_part] of integer;       {school code for student}

   sch_name:array [1..max_sch] of string[50];           {school name}

   stu_code:array [1..max_part] of string[7];           {student code}

   stu_code_s:array [1..max_part] of string[7];        {student code for sorting}

   stu_name:array [1..max_part] of string[15];          {student name}

   stu_ans:array [1..max_question] of string[max_question+1];    {mc answer of student}
```

```
ori_ans:string;                                      {model answer of mc}

p,p1,p2:integer;                                     {variable for school info}

s,s1,s2:integer;                                     {variable for student info and ans}

a,a1:integer;                                        {variable for answer key}

c:integer;                                           {variable for counting}

num_sch,num_stu:integer;                             {Number of schools and students}

Sa,Ori:string;                                       {Vaviable of cheking answer}

marks:array [1..max_part] of integer;                {Marks count for stuent}

marks_s:array [1..max_part] of integer;              {Marks for sorting}

place:array [1..max_part] of integer;                {Order of sorting}

Q_marks:array [1..max_question] of integer;          {Marks for calculating percentage}

Q_percen:array [1..max_question] of real;            {Percentage correct of each question}

i,j,x,y,z:integer;                                   {Array variable}

freq:array [1..max_part] of integer;                 {For calculate number of participants for each school}

sch_marks:array [1..max_sch] of integer;             {For calculate total marks of each school}

ave_marks:array [1..max_sch] of real;                {For calculate average marks of each school}

chart_sch_marks:array [1..max_sch] of integer;       {Value of average marks in chart}
```

Constant can change or modify according to the information needed to be input. If the constant is changed, the whole program will modify to optimum state.

Variable will change during processing of program, the variable was not design to be change be user.

As everyone knows, the first step to construct this program is read in data, therefore the first main procedure is to let user input their data file. The program code of this part of procedure will show below.

Read-in Data:

```
Procedure Read_data; {read participants' info}

Var
    sch_info:text;    {info of schools}
    stu_info:text;    {info of participants and their ans}
    ans_key:text;      {answer key of mc}
    filename1,filename2,filename3:string; {filename of text file}

  Begin
       writeln('Please enter the filename which contain following information.');
       write('Information of schools:');
       readln(filename1);
       assign(sch_info,filename1);
       reset(sch_info);
       p:=0;
       while not eof(sch_info) do
               begin
                      p:=p+1;
                      readln(sch_info,sch_code[p],sch_name[p]);
               end;
```

In this procedure, three special function, assign(), reset() and eof() are used. These three functions are essential for reading files.

The next procedure is about the data processing, and it is the answer checking process, the algorithms is simple in this procedure. A nested for-loop is introduced into this procedure. the student code is read first then to the question number. After that the model from answer key will be compared with student answer using the function, copy(), then the marks of student will be calculated and save into array.

The code of answer checking procedure will shown below:

```
Procedure answer_checker;    {check students answer}
```

```
  Begin
    for x:=1 to s do
      begin
        marks[x]:=0;
        for y:=1 to max_question do
          begin
            Ori:=copy(ori_ans,y,1);
            Sa:=copy(stu_ans[x],y+1,1);
            if ori=Sa
              then marks[x]:=marks[x]+1;
          end;
       end;
  end;
```

The next procedure is correct percentage checking which is almost as same as answer checking procedure. however, the only different between is the reverse order of nested for-loop, the order of questions and student reverse, so question will be consider first, then the student answer. So the correct rate of each question will be recorded, and divided by the number of total participants, the percentages are then obtained and save into a array.

After percentage and marks checking, the procedure I wrote is sorting of students' marks. This procedure is crucial if I want to give user the result of winner.

So as to produce this procedure, I used nested for-loop again. However, this kind of nested for-loop is tricky, therefore I asked my CIT teacher for help. Finally this step can be carried out without error.

The code of Sorting is shown below:

```
Procedure Sort_stu_marks;      {For sorting}
var temp_stumark:integer;
    temp_stucode:string;
  Begin
    for x:=1 to s do
      begin
       marks_s[x]:=marks[x];
```

```
          stu_code_s[x]:=stu_code[x];
        end;
            for z:=1 to s-1 do
                for    x:=1 to s-z do
                    if marks_s[x]<marks_s[x+1] then
                        begin
                            temp_stumark:=marks_s[x];
                            marks_s[x]:=marks_s[x+1];
                            marks_s[x+1]:=temp_stumark;
                            temp_stucode:=stu_code_s[x];
                            stu_code_s[x]:=stu_code_s[x+1];
                            stu_code_s[x+1]:=temp_stucode;
                        end;
End;
```

The code shown clearly, that a nested for-loop is used, there are few local variable which are used for sorting only. This kind of variable will not effect the global programming environment nor use it other procedure.

Some minor data such as total number of participants and schools also needed to be considering in a procedure. The procedures are very simple as the data is store in a global array. Data just need to extract from the array and write out or write into new variable.

Here is the procedure of counting:

```
Procedure Count_sch_part; {Count the number of participants and schools}

    Begin
            for c := 1 to max_part do
            begin
               if (sch_code_stu[c]<>sch_code_stu[c-1]) and (sch_code_stu[c]>0)
                   then num_sch:=num_sch+1;
               if sch_code_stu[c]>0
                   then num_stu:=num_stu+1;
            end;
End;
```

However, there is another procedure, which is a bit tricky, this procedure is the counting of number students from each school. This program used a nested for-loop to count the number of students from each school.

The trickiest part is this procedure, is that the variable for storing school code is an integer, so the number of students can be counted by compare two school codes. Therefore, the number of students can be counted correctly even their school code order is randomized.

The procedure of counting students from each school is shown below:

```
Procedure Total_stu_sch;      {Calculate the total number of students form each school}

   Begin
      for p1:=1 to p do
         for s1:=1 to s do
            begin
               if sch_code_stu[s1]=sch_code[p1] then
                  freq[p1]:=freq[p1]+1;
            end;
   End;
```

In this procedure, a global array, freq, is introduced into this procedure. It is used for counting. However, two-dimensional also can be used for counting students from each school, it is even more efficient and can avoid to use this procedure, but the advantages is that setting up a two-dimensional array is complicated than one-dimensional.

One of the important procedure in this program, is calculating the school average marks. This procedure is quite simple. However, it required a logical comparison to avoid runtime error and sudden termination of program.

Therefore, there is one school in the example file, which has only the school name but don’t contain any participant. While reading the data file, the program will avoid counting this school as no one participated in it.

There are two procedure related to chart outputting. The reason that I used two procedures instead of one is to avoid overwriting or rewriting of variable. The result will be double-counted, if the user run the same procedure in the menu twice.

The procedure of chart:

```
Procedure Pre_chart;               {Calculate the value for chart}

   Begin
      for p1:=1 to p do
         begin
            if freq[p1]<>0 then
               chart_sch_marks[p1]:=sch_marks[p1] div freq[p1];
         end;
   End;

Procedure Chart_sch;               {Chart for school average marks}

   Begin
      writeln('The following chart show the average marks of each school');
      writeln('School code    Marks');
      for p1:=1 to p do
         begin
            write(sch_code[p1]);
            write('                    ');
            for p2:=1 to chart_sch_marks[p1] do
               begin
                  write('~');
               end;
            writeln;
         end;
   End;
```

As shown above, this procedure is split into two parts, this provide convenience for me to calculate the result and output result respectively. This prevent no data output or wrong data output in chart. Moreover, in procedure (pre_chart), I used a function div, this turn the real type school average marks into an integer. I used (div 1) as it will round off whatever the numbers are.

After the sorting procedure, there is procedure which can list out the rank of all participants. It is one of the procedures of output.

Here is the code of listing:

```
Procedure List_place;      {List the order of student according to marks}

  Begin
    writeln('First    Student number');
    writeln('1':4,stu_code_s[1]:11);
    writeln('Second    Student number');
    writeln('2':4,stu_code_s[2]:11);
    writeln('First    Student number');
    writeln('3':4,stu_code_s[3]:11);
    writeln('Place    Student number');
    for x:=4 to s do
      writeln(x:4,stu_code_s[x]:11);
  End;
```

In the end of the program, there is a part which is the main program part, is consist of the procedure which needed to calculate result and input data, and a procedure of menu, which allows user to choose any information to be displayed.

Any further explain of algorithms and functions in procedure or program will be discussed in chapter 4.3.

4.2 Data Structures

As saw in the variable list in chapter 4.1, most of you may noticed many of the variable is in the form of array. The reason that a use array is simple, array can store a set of data in a single variable with respect to a integer in array. As long as the integer does not exceed to maximum value of array, data can still be stored.

I use one- dimensional array as mentioned in chapter 4.1, the pros of one-dimensional array over that of two-dimensional array is simplicity. The data need not to be well organized as if I use two-dimensional array. However the data stored in two-dimensional array is easier to handle.

But for my convenience in handling and modifying my program and data, I choose to use one-dimensional array.

The array will be listed as below:

```
sch_code:array [1..max_sch] of integer;                {school code}
sch_code_stu:array [1..max_part] of integer;        {school code for student}
sch_name:array [1..max_sch] of string[50];            {school name}
stu_code:array [1..max_part] of string[7];            {student code}
stu_code_s:array [1..max_part] of string[7];        {student code for sorting}
stu_name:array [1..max_part] of string[15];          {student name}
stu_ans:array [1..max_question] of string[max_question+1];    {mc answer of student}
```

These are part of the list of array, it store the information of participants

Eg. 1

Sch_code[1],Sch_name[1]

Store school code and school name of the first school

eg. 2

Stu_code[1],Stu_name[1]Stu_ans[1]

Store student code, student name and student answer respectively.

4.3 Procedures in the Program

In my actual construction of the program, there are a bit different from my original design.

The main different is about inputting method, I introduce a auto-input method, which allow user to press a single key to enter all the data needed to be processed, the only restriction of this method is that the filename cannot be modified. Or data will be inputted incorrectly.

I enjoy to use this feature, as it reduce time for debugging the program as I need to input less things.

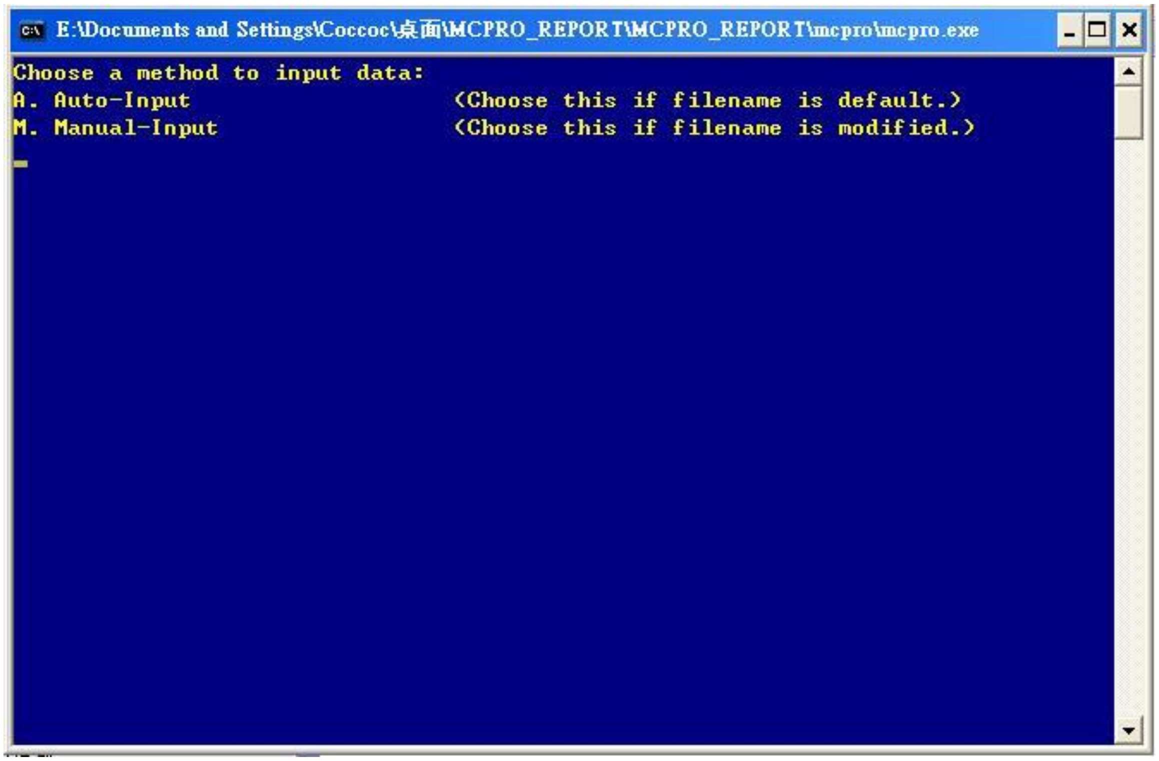
The code of input interface is shown below:

```
Procedure Met_read;    {Choose method for read in data}
Var
  met:char;
  Begin
    writeln('Choose a method to input data:');
    writeln('A. Auto-Input                         (Choose this if filename is default.)');
    writeln('M. Manual-Input                        (Choose this if filename is modified.)');
    readln(met);
    case met of
    'a','A':Auto_read;
    'm','M':Manual_read;
    else
      begin
        writeln;
        writeln('*************************************');
        writeln('***********Invalid output************');
        writeln('******Auto return in one second******');
        writeln('*************************************');
        delay(1000);
        clrscr;
        met_read;
      end;
    end
```

```
end;
```

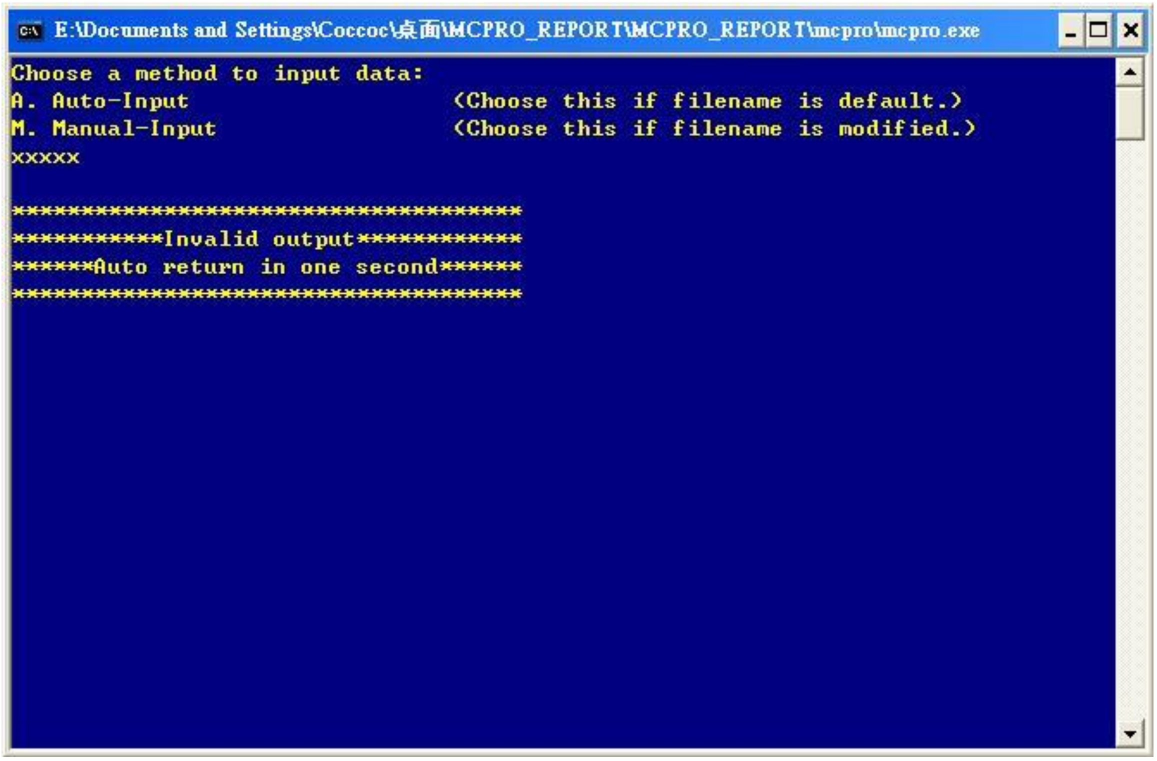
The method that I change manual-input into auto-input is simple, I just fix the filename into a const, and every time the file can be input correctly as long as it is in the same folder with the program.

Some of the screen of the input interface is shown below:


E:\Documents and Settings\Coccoc\桌面\MCPRO_REPORT\MCPRO_REPORT\mcpro\mcpro.exe

```
Choose a method to input data:
A. Auto-Input                       (Choose this if filename is default.)
M. Manual-Input                     (Choose this if filename is modified.)
```

This is the initial screen of data input.

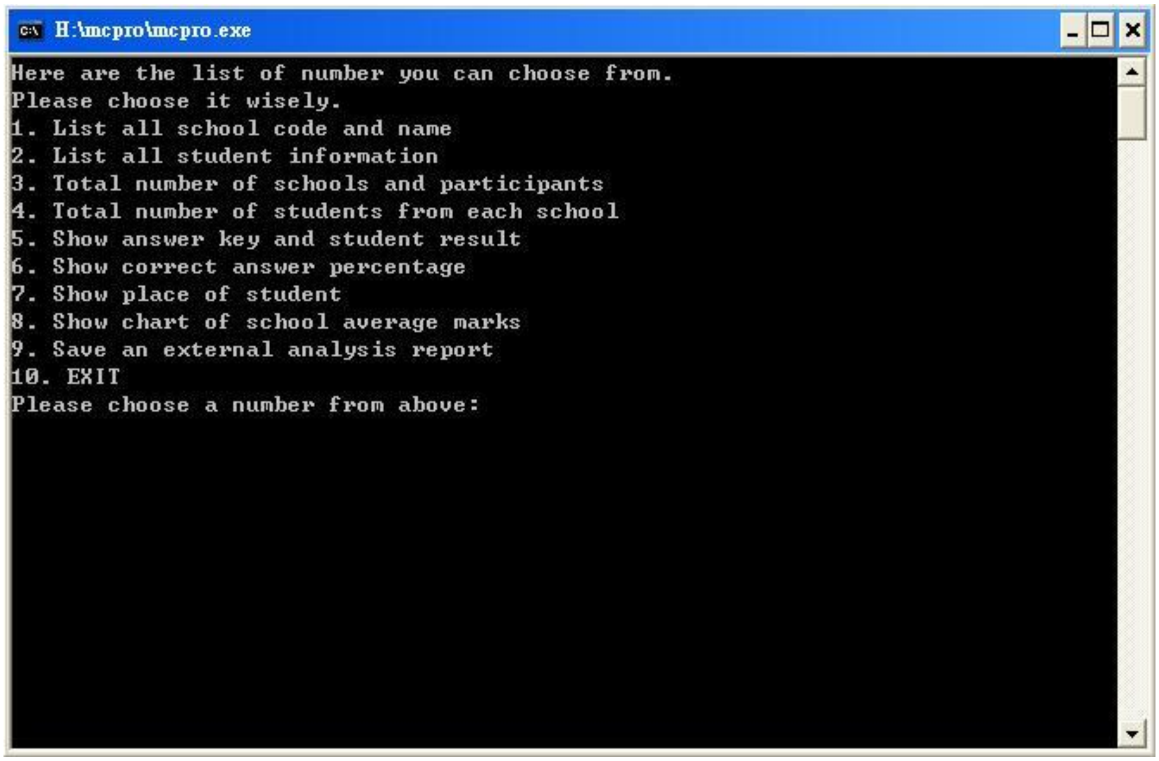

E:\Documents and Settings\Coccoc\桌面\MCPRO_REPORT\MCPRO_REPORT\mcpro\mcpro.exe

```
Choose a method to input data:
A. Auto-Input                   (Choose this if filename is default.)
M. Manual-Input                 (Choose this if filename is modified.)
xxxxx

*************************************
***********Invalid output************
******Auto return in one second******
*************************************
```

This is the screen shown wrong choice is inputted.

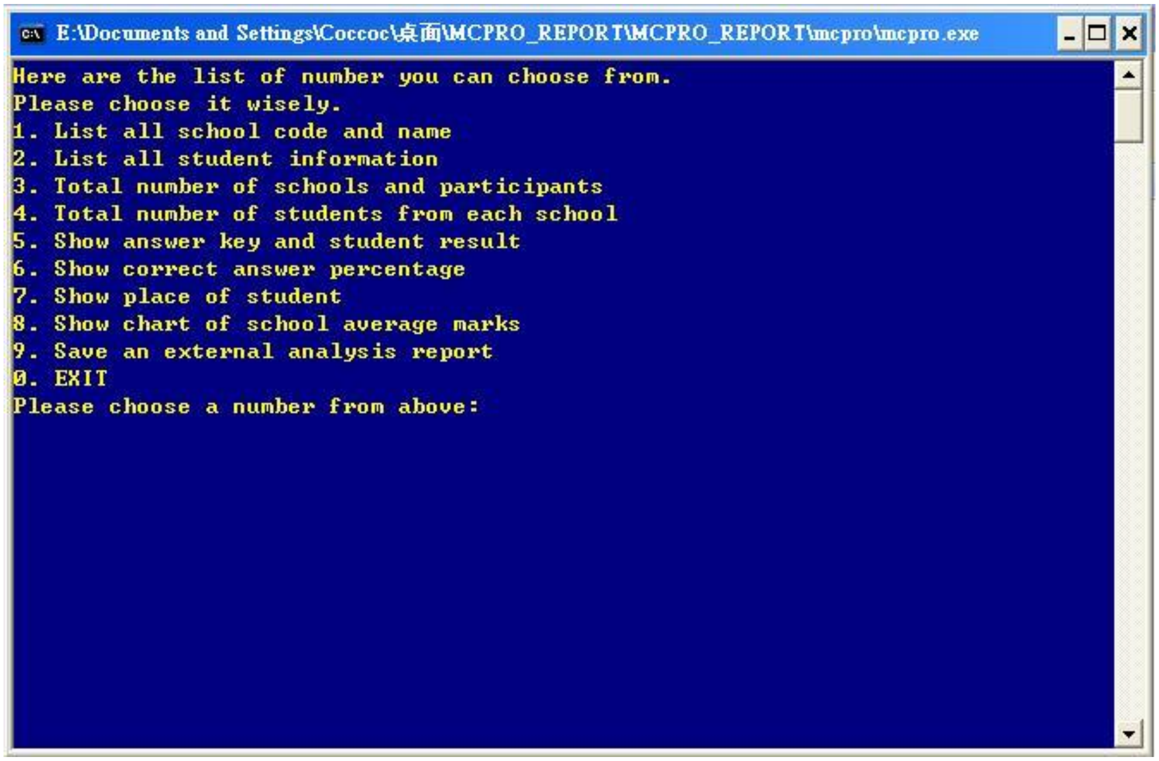
The other feature that differs is the text colour and background colour. The text colour and background colour are originally white and black respectively. But I found that this kind of combination is not good for user to read, so I change it to a shape and contrast colour.

Here are the comparison between old colour and new colour:



```
Here are the list of number you can choose from.
Please choose it wisely.
1. List all school code and name
2. List all student information
3. Total number of schools and participants
4. Total number of students from each school
5. Show answer key and student result
6. Show correct answer percentage
7. Show place of student
8. Show chart of school average marks
9. Save an external analysis report
10. EXIT
Please choose a number from above:
```

This is the original colour.


E:\Documents and Settings\Coccoc\桌面\MCPRO_REPORT\MCPRO_REPORT\mcpro\mcpro.exe

```
Here are the list of number you can choose from.
Please choose it wisely.
1. List all school code and name
2. List all student information
3. Total number of schools and participants
4. Total number of students from each school
5. Show answer key and student result
6. Show correct answer percentage
7. Show place of student
8. Show chart of school average marks
9. Save an external analysis report
0. EXIT
Please choose a number from above:
```

This is the new colour.

As shown as above, the different between two colour are significant, and obviously the new one is better and have higher readability.

About other procedure and function, most of them are already listed and described in chapter 4.1, so I am going focus on the output part only.

The output is simple, as the variables are ready to be outputted, the only things that I should concern is only the output format.

Therefore, let me explain the format with the aid of program code:

```
Procedure Output_report;     {To save the report in a text file}
var
    name:string;
    report:text;
  Begin
    writeln('Please give a name to the output file:');
    readln(name);
    name:=name+'.txt';
    assign(report,name);
    rewrite(report);
      begin
        writeln(report,'1.The total number of schools & students');
       writeln(report,'****************************************************************');
        writeln(report,'Total number of school(s):',num_sch,'    Total number of participant(s):',num_stu);
        writeln(report);

        writeln(report,'2.The information of school');
       writeln(report,'****************************************************************');
        writeln(report,'School Code              School Name');
          for p1:=1 to p do
              writeln(report,sch_code[p1]:6,sch_name[p1]:23);
        writeln(report);

        writeln(report,'3.The information of participants');
writeln(report,'***************************************************************');
 writeln(report,'School Code             School Name           Student Code           Student Name');
     for s1:=1 to s do
       begin
         s2:=sch_code_stu[s1];
         writeln(report,sch_code_stu[s1]:6,sch_name[s2]:23,stu_code[s1]:12,stu_name[s1]:27)
       end;
 writeln(report);

 writeln(report,'4.Total number of students from each school');
writeln(report,'***************************************************************');
 writeln(report,'School code    School name    Number of participants    Average marks');
     for p1:=1 to p do
         begin
             writeln(report,sch_code[p1]:6,sch_name[p1]:18,freq[p1]:10,ave_marks[p1]:25:3);
         end;
 writeln(report);

 writeln(report,'5.Answer key & Student result');
writeln(report,'***************************************************************');
 writeln(report,'Answer key');
 writeln(report,ori_ans);
 writeln(report,'student number    Number of correct    Answer of students');
     for x:=1 to s do
         begin
             writeln(report,x:7,marks[x]:17,stu_ans[x]:Max_question+11);
         end;
 writeln(report);

 writeln(report,'6.Correct answer percentage');
writeln(report,'***************************************************************');
 writeln(report,'Question No.    No.(s) of Correct    Correct percentage');
     for y:=1 to max_question do
         begin
             writeln(report,y:11,Q_marks[y]:14,Q_percen[y]:19:3);
         end;
 writeln(report);
```

```
      writeln(report,'7.Show place of student');
      writeln(report,'********************************************************************');
      writeln(report,'First    Student number');
      writeln(report,'1':4,stu_code_s[1]:11);
      writeln(report,'Second    Student number');
      writeln(report,'2':4,stu_code_s[2]:11);
      writeln(report,'First    Student number');
      writeln(report,'3':4,stu_code_s[3]:11);
      writeln(report,'Place    Student number');
        for x:=4 to s do
          writeln(report,x:4,stu_code_s[x]:11);
      writeln(report);

      writeln(report,'8.Chart showing school average marks');
      writeln(report,'********************************************************************');
      writeln(report,'The following chart show the average marks of each school');
      writeln(report,'School code    Marks');
        for p1:=1 to p do
          begin
            write(report,sch_code[p1]);
            write(report,'                           ');
            for p2:=1 to chart_sch_marks[p1] do
              begin
                write(report,'*');
              end;
            write(report,'    ',chart_sch_marks[p1]);
            writeln(report);
          end;
      writeln(report);



    end;
  close(report);
End;
```

This is the full procedure of output report, the reason that quote this for explain is because the output report format is more or lee the same as the one on-screen. As you can , see from the code, each information is listed out by writeln. These is possible because all information is gathered and calculated, so the output is easy as it is required to list out variable and array only.

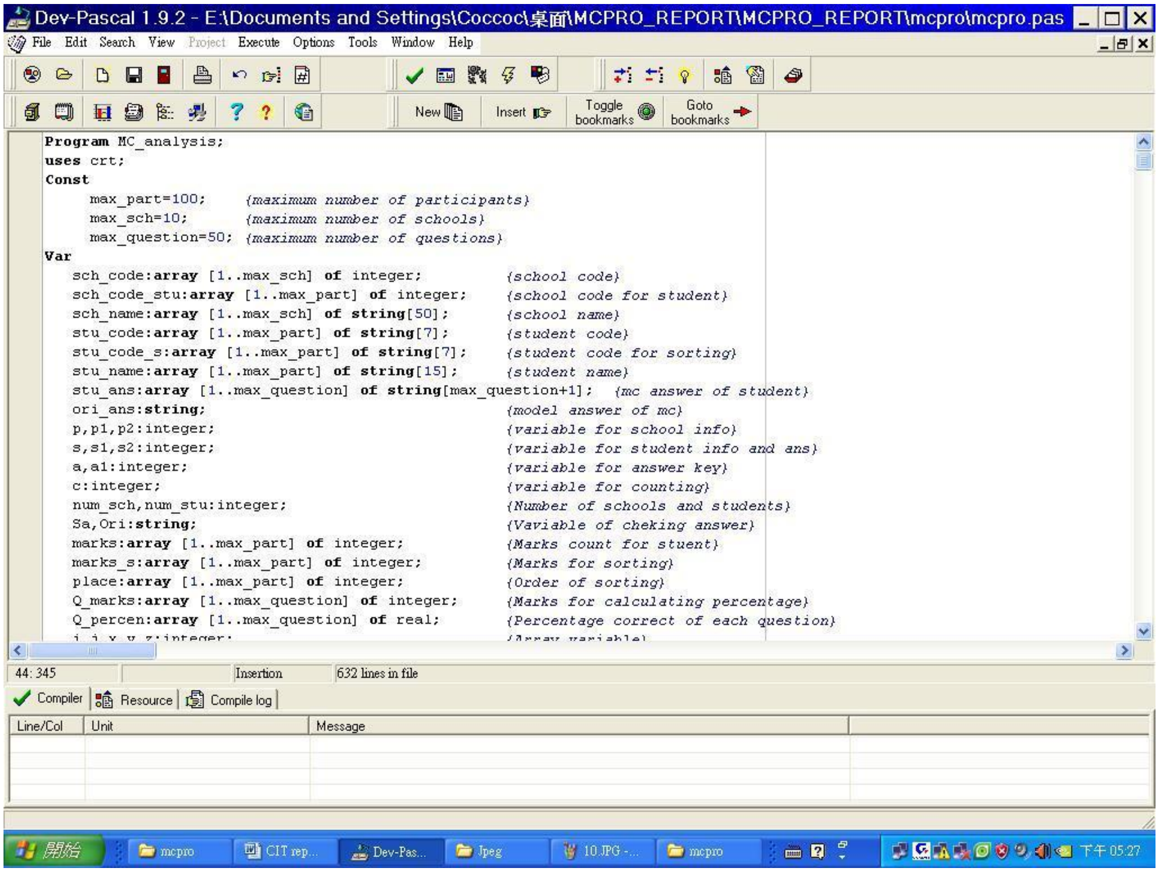
4.4 Program Coding

I used Dev Pascal version 1.9.2 through out my project, it has an advantage over QPascal, the user interface, it has a better and clearer user interface than QPascal, so I can view and modify the code more efficiently..

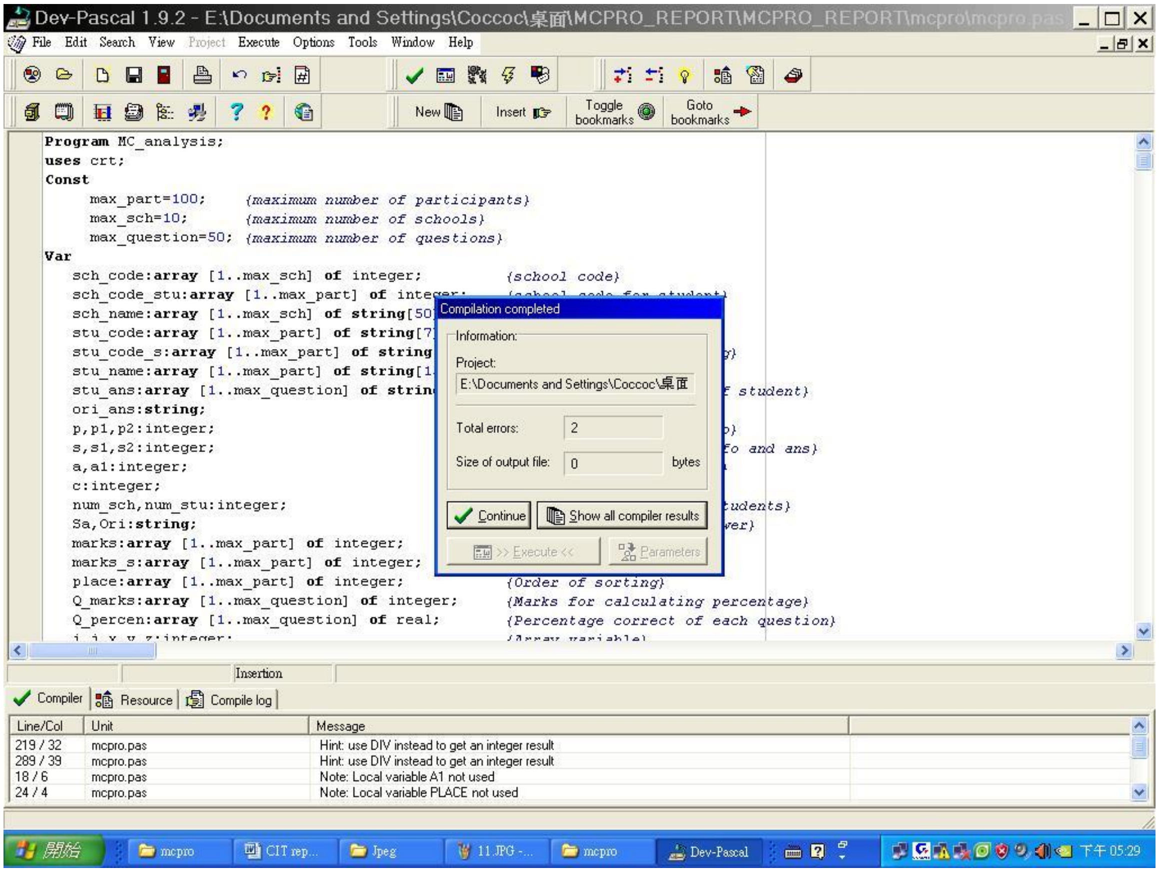
I saved My source code as mcpro.pas together with the executable mcpro.exe.

The full version of program code will be in appendix.

Here are some screen shots of coding.



```
Program MC_analysis;
uses crt;
Const
     max_part=100;    {maximum number of participants}
     max_sch=10;      {maximum number of schools}
     max_question=50; {maximum number of questions}
Var
   sch_code:array [1..max_sch] of integer;         {school code}
   sch_code_stu:array [1..max_part] of integer;    {school code for student}
   sch_name:array [1..max_sch] of string[50];      {school name}
   stu_code:array [1..max_part] of string[7];      {student code}
   stu_code_s:array [1..max_part] of string[7];    {student code for sorting}
   stu_name:array [1..max_part] of string[15];     {student name}
   stu_ans:array [1..max_question] of string[max_question+1];  {mc answer of student}
   ori_ans:string;                                 {model answer of mc}
   p,p1,p2:integer;                                {variable for school info}
   s,s1,s2:integer;                                {variable for student info and ans}
   a,a1:integer;                                   {variable for answer key}
   c:integer;                                      {variable for counting}
   num_sch,num_stu:integer;                        {Number of schools and students}
   Sa,Ori:string;                                  {Vaviable of cheking answer}
   marks:array [1..max_part] of integer;           {Marks count for stuent}
   marks_s:array [1..max_part] of integer;         {Marks for sorting}
   place:array [1..max_part] of integer;           {Order of sorting}
   Q_marks:array [1..max_question] of integer;     {Marks for calculating percentage}
   Q_percen:array [1..max_question] of real;       {Percentage correct of each question}
   i,j,x,y,z:integer;                              {Array variable}
```



Dev-Pascal 1.9.2 - E:\Documents and Settings\Coccoc\桌面\MCPRO_REPORT\MCPRO_REPORT\mcpro\mcpro.pas

```
Program MC_analysis;
uses crt;
Const
     max_part=100;     {maximum number of participants}
     max_sch=10;       {maximum number of schools}
     max_question=50;  {maximum number of questions}
Var
   sch_code:array [1..max_sch] of integer;          {school code}
   sch_code_stu:array [1..max_part] of integer;     {school code for student}
   sch_name:array [1..max_sch] of string[50
   stu_code:array [1..max_part] of string[7
   stu_code_s:array [1..max_part] of string                                   g}
   stu_name:array [1..max_part] of string[1
   stu_ans:array [1..max_question] of strin                                   f student}
   ori_ans:string;
   p,p1,p2:integer;                                                           p}
   s,s1,s2:integer;                                                           fo and ans}
   a,a1:integer;
   c:integer;
   num_sch,num_stu:integer;                                                   tudents}
   Sa,Ori:string;                                                             wer}
   marks:array [1..max_part] of integer;
   marks_s:array [1..max_part] of integer;
   place:array [1..max_part] of integer;            {Order of sorting}
   Q_marks:array [1..max_question] of integer;      {Marks for calculating percentage}
   Q_percen:array [1..max_question] of real;        {Percentage correct of each question}
   i,j,x,y,z:integer;                               {Array variable}
```

Compilation completed

Information:

Project: E:\Documents and Settings\Coccoc\桌面

Total errors: 2

Size of output file: 0 bytes

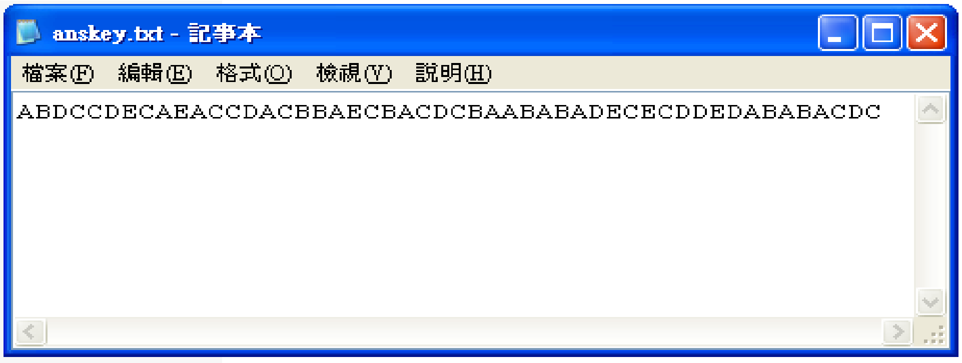
```
Line/Col  Unit       Message
219 / 32  mcpro.pas  Hint: use DIV instead to get an integer result
289 / 39  mcpro.pas  Hint: use DIV instead to get an integer result
18 / 6    mcpro.pas  Note: Local variable A1 not used
24 / 4    mcpro.pas  Note: Local variable PLACE not used
```

4.5 Program Execution

Some files should be executed before running the program, as mention above, three text file containing information should be executed first.

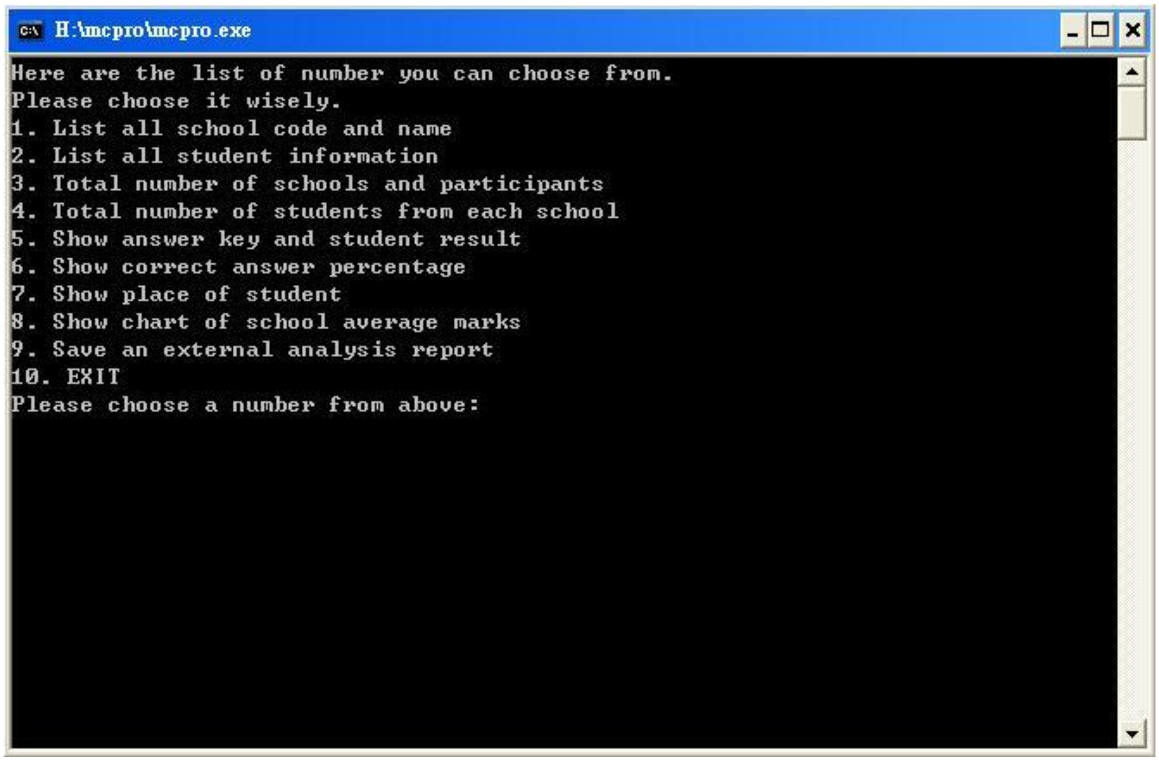
That the anskey.txt, stu_ans,txt and sch_info,txt.

Also, a report will be produced after running the program. However the filename can be defined by user. To provide maximum convenience, .txt need not to be included.



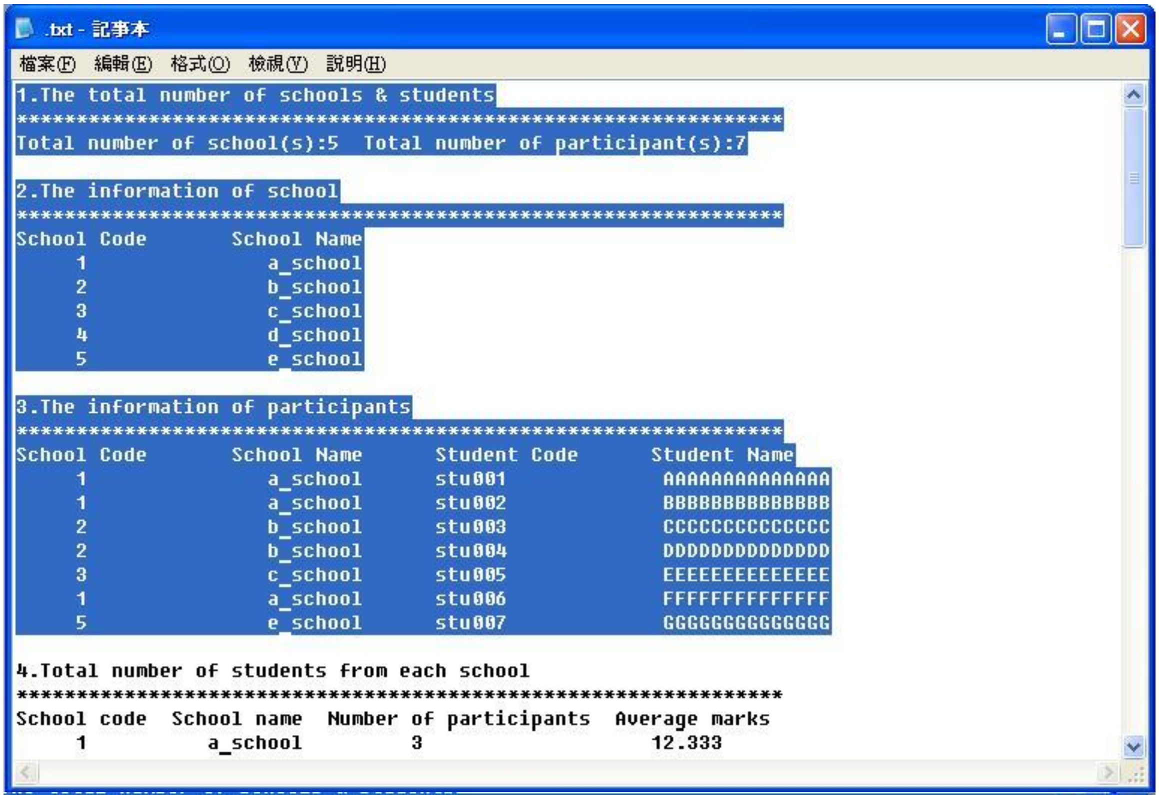
```
ABDCCDECAEACCDACBBAECBACDCBAABABADECECDDEDABABACDC
```

User interface of the program (Menu)

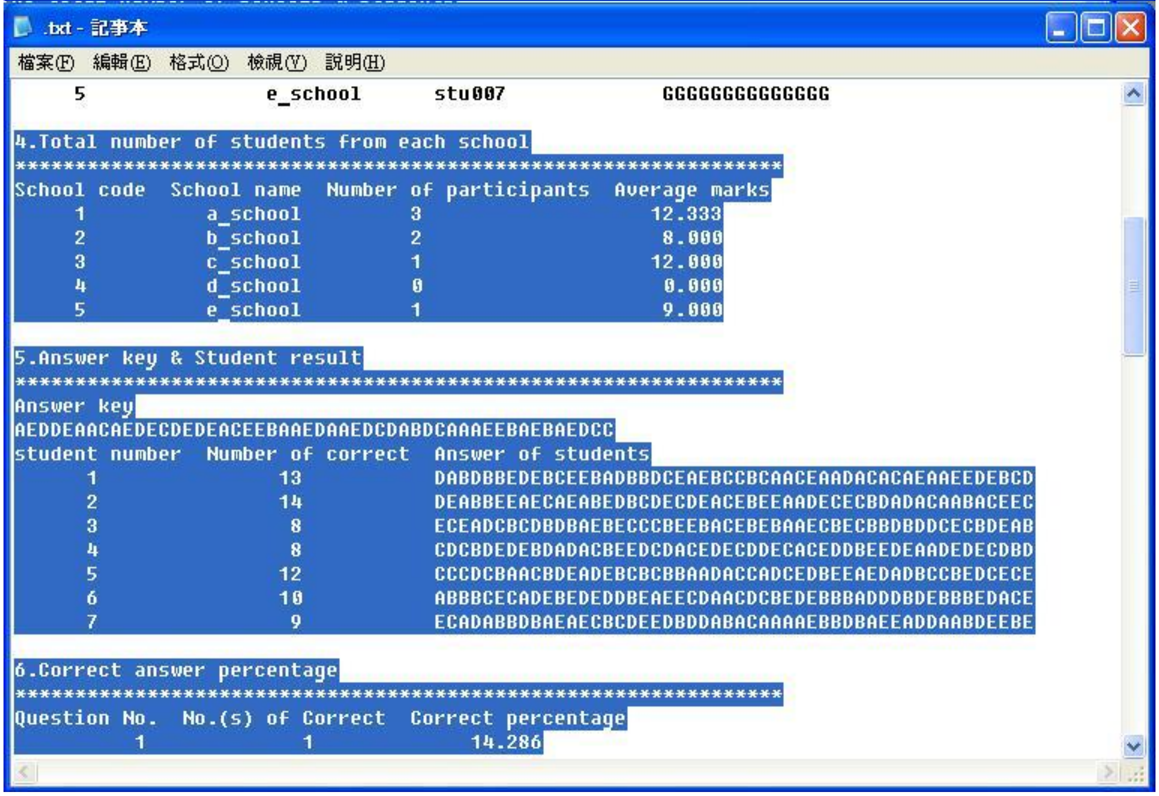


```
Here are the list of number you can choose from.
Please choose it wisely.
1. List all school code and name
2. List all student information
3. Total number of schools and participants
4. Total number of students from each school
5. Show answer key and student result
6. Show correct answer percentage
7. Show place of student
8. Show chart of school average marks
9. Save an external analysis report
10. EXIT
Please choose a number from above:
```

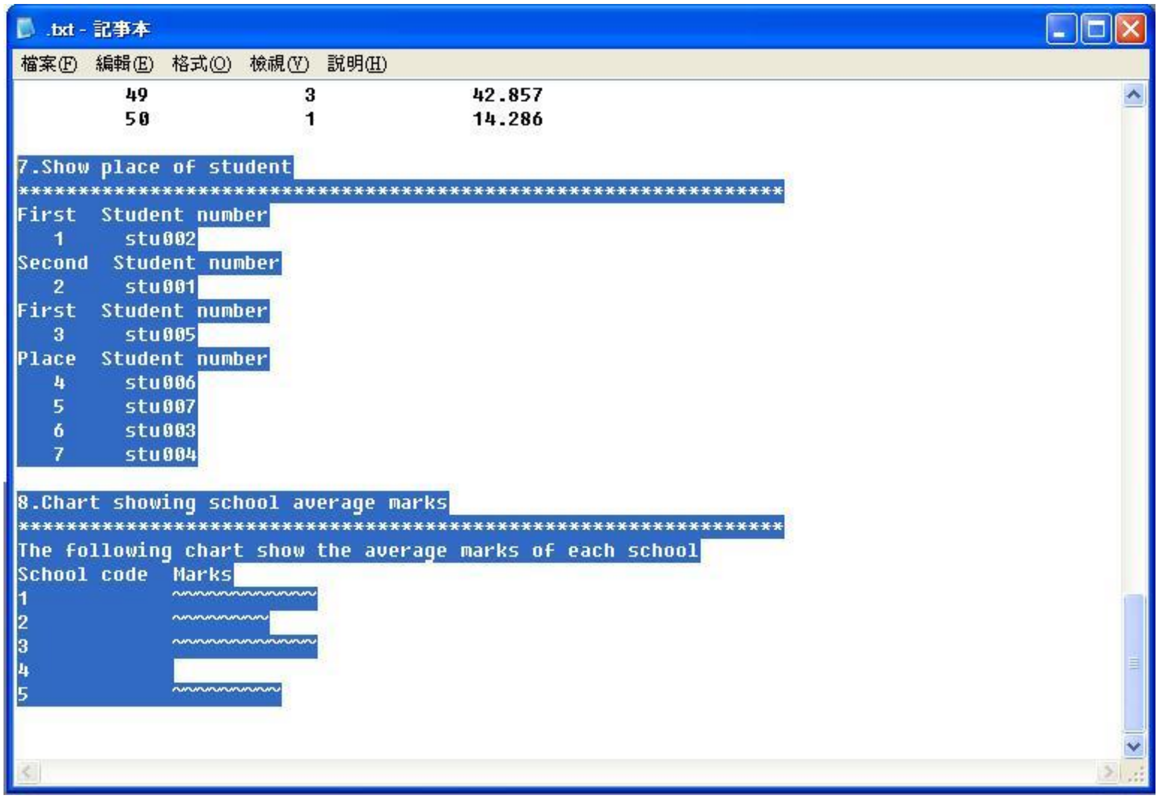
Results displayed in text file



```
1.The total number of schools & students
****************************************************************
Total number of school(s):5  Total number of participant(s):7

2.The information of school
****************************************************************
School Code       School Name
     1               a_school
     2               b_school
     3               c_school
     4               d_school
     5               e_school

3.The information of participants
****************************************************************
School Code       School Name      Student Code      Student Name
     1               a_school      stu001             AAAAAAAAAAAAAA
     1               a_school      stu002             BBBBBBBBBBBBBB
     2               b_school      stu003             CCCCCCCCCCCCCC
     2               b_school      stu004             DDDDDDDDDDDDDD
     3               c_school      stu005             EEEEEEEEEEEEEE
     1               a_school      stu006             FFFFFFFFFFFFFF
     5               e_school      stu007             GGGGGGGGGGGGGG

4.Total number of students from each school
****************************************************************
School code  School name  Number of participants  Average marks
     1          a_school         3                      12.333
```



```
     5               e_school      stu007             GGGGGGGGGGGGGG

4.Total number of students from each school
****************************************************************
School code  School name  Number of participants  Average marks
     1          a_school         3                      12.333
     2          b_school         2                       8.000
     3          c_school         1                      12.000
     4          d_school         0                       0.000
     5          e_school         1                       9.000

5.Answer key & Student result
****************************************************************
Answer key
AEDDEAACAEDECDEDEACEEBAAEDAAEDCDABDCAAAEEBAEBAEDCC
student number  Number of correct  Answer of students
      1                13          DABDBBEDEBCEEBADBBDCEAEBCCBCAACEAADACACAEAAEEDEBCD
      2                14          DEABBEEAECAEABEDBCDECDEACEBEEAADECECBDADACAABACEEC
      3                 8          ECEADCBCDBDBAEBECCCBEEBACEBEBAAECBECBBDBDDCECBDEAB
      4                 8          CDCBDEDEBDADACBEEDCDACEDECDDECACEDDBEEDEAADEDECDBD
      5                12          CCCDCBAACBDEADEBCBCBBAADACCADCEDBEEAEDADBCCBEDCECE
      6                10          ABBBCECADEBEDEDDBEAEECDAACDCBEDEBBBADDDBDEBBBEDACE
      7                 9          ECADABBDBAEAECBCDEEDBDDABACAAAAEBBDBAEEADDAABDEEBE

6.Correct answer percentage
****************************************************************
Question No.  No.(s) of Correct  Correct percentage
          1             1             14.286
```



```
         49             3             42.857
         50             1             14.286

7.Show place of student
****************************************************************
First  Student number
   1     stu002
Second  Student number
   2     stu001
First  Student number
   3     stu005
Place  Student number
   4     stu006
   5     stu007
   6     stu003
   7     stu004

8.Chart showing school average marks
****************************************************************
The following chart show the average marks of each school
School code  Marks
1            ~~~~~~~~~~~~
2            ~~~~~~~~
3            ~~~~~~~~~~~~
4            
5            ~~~~~~~~~
```

Chapter 5 Testing & Evaluation

5.1 Internal testing

In my program coding process, I debug when a started to write a new procedure. However, I spot out some error eventually.

Here are the main bugs and test I encountered and carried.

Test case 1.1

Purpose:	To check if input data can be shown correctly
Input:	Normal answer keys file and participants' answers file. The files are dragged into the program
Expected Output:	The data in the file are outputted on screen
Actual Output:	Runtime error occurs
Test Result:	Fail
Follow-up Action:	Check the read in data procedure.

Test case 1.2

Purpose:	Follow-up for test 1.1
Input:	Normal answer keys file and participants' answers file. Type in the file name.
Expected Output:	The data in file are outputted on screen.
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass
Follow-up Action:	Nil

I then found that the bugs occur in test 1.1 is due to the Chinese character in the file path, I the n modified the file path, so drag and type-in file name can also be use.

Test case 2.1

Purpose:	To check the calculated marks of students
Input:	Compare the marks calculated by the program and my manually calculated marks.
Expected Output:	Two marks should also be the same value
Actual Output:	All actual results are the same as the expected results.
Test Result:	Pass / No bugs found
Follow-up Action:	Nil

Test case 3

Purpose:	To check the correctness of percentage outputted
Input:	Run the correct percentage calculating procedure
Expected Output:	A real type percentage should be outputted
Actual Output:	A correct percentage is outputted, however the significant figure is too large.
Test Result:	Merely pass.
Follow-up Action:	Reduce the significant figure into one decimal place.

Test case 4

Purpose:	To check if the graph can be successfully outputted
Input:	Run the write out graph procedure
Expected Output:	A horizontal graph produced.
Actual Output:	The graph symbol '~' loop continuously.
Test Result:	Fail
Follow-up Action:	Correct the variable in the for-loop of graph procedure.

5.2 Self-Evaluation

Pros and features:

1. Most function and data can be input automatically, as I am featuring the easiest way to find out result of this competition, so I introduce a procedure which help user to enter the file. It is very convenient, and it does not appear in other students' work. (as a see from their published work.)
2. Simplified menu provide a handy and clear screen for user.
3. High readability as I introduce a contrasted background and text colour, which is clearer than the original background and text colour.
4. User-friendly and fun user interface.
5. High flexibility for further development.
6. No error may occur when saving report, even the user didn't enter anything.

Cons:

1. Functions are limited as I want to show only the important information, so there is few extra options for user to choose from.
2. There is a know bug, that the rank will be slightly if two or more participants score the same marks.

5.3 Feedback and external evaluation

I keen on receiving feedback from other student, however only some of them reply.

Here is a comment from my classmate Gary SO:

1, The use of '1010' in the beginning is a feeling of statistics and messy.
It can't be shown with blue background and yellow text.
2. No need to show answer key when there is so many questions
3. Run-time error is occurred when letters is tpyed on the front page.
4. The " ~ " symbols used in the graph were hard to count and the exact mark of schools cannot be shown.
5. Modify the data file if you like.

The above comment didn’t undergo any modification.(even typing mistakes)

Some of the bugs he stated, such as no.2 & 3, is useful for improving my program, so I took his idea and change the ‘~’ symbol in the graph to ‘*’. Moreover, I added the numerical marks besides the graph.

Secondly, I modified the menu system, and change the integer type variable to character type, so this can avoid runtime error in the menu if user input wrong choice.

Some other classmate also gave me comment orally, so let me mention it below.

1. **The program should contain a menu.**
2. **The result shouldn’t output together in one page.**
3. **Runtime error may occur when inputting files.**

I Learned their comments, I then correct no.1 by making a menu which allows user to choose from.

The second bugs is corrected together with no.1, as the result is divided by the menu,

Running error is still easy to occur if user drag the file into the program, so I introduce auto-input system to avoid errors and provide convenience.

Chapter 6 Conclusion & Discussion

6.1 Pros and cons of my Program

The pros and cons is summarize in chapter 5.2, but I still want to mention the target of my program. It is to maximize the convenience. Nevertheless, I successfully achieve my aim. And foreseeable, not too much classmate appeal to achieve this aim as I did.

However, the things that I lack of are extra function and decoration. So my program may not be as useful as someone, and appear dull.

6.2 Future Improvement

What I needed to improve are simple, generally, I have three main ideas to improve my program.

But, before improvement, the most important bugs that I have to fix is the rank procedure, I also found this bugs occur in my classmates' program.

Firstly, I am going to decorate my program just the others did, however, I will decorate it uniquely.

Secondly, I will add more function to my program, such as calculating the mean, mode, medium and standard deviation.

Finally, my ultimate goal is to create a single-key program, which enables the user to press only one key, and then all data and result will be process and output. This is what I called The Maximum convenience.

6.3 Self-Reflection

After doing this programming project, I found that I got a better understanding on the underlying principle of program, and knowing the programming world deeper.

I keen on to investigate and develop new and useful programs if I had leisure time after HKCEE. And hopefully, these programming techniques will be useful in my future daily life.

Chapter 7 Reference and Acknowledgement

From Internet websites

1. http://www.wikipedia.org
2. http://en.wikipedia.org/wiki/Pascal_(programming_language)
3. http://en.wikipedia.org/wiki/BASIC

From books

1. 2009 Computer and information technology coursework : Skills & practice Module A

2. Pascal programming language notes from CSWCSS

Acknowledgement

1. CIT teacher MR. Chu

2. Classmate TseCP

3. Classmate SoGW

Appendices

Appendix 1 – Program Code

```
Program MC_analysis;
uses crt;
Const
        max_part=100;       {maximum number of participants}
        max_sch=10;          {maximum number of schools}
        max_question=50; {maximum number of questions}
Var
    sch_code:array [1..max_sch] of integer;                {school code}
    sch_code_stu:array [1..max_part] of integer;       {school code for student}
    sch_name:array [1..max_sch] of string[50];           {school name}
    stu_code:array [1..max_part] of string[7];           {student code}
    stu_code_s:array [1..max_part] of string[7];        {student code for sorting}
    stu_name:array [1..max_part] of string[15];         {student name}
    stu_ans:array [1..max_question] of string[max_question+1];     {mc answer of
student}
    ori_ans:string;                                                     {model answer of mc}
    p,p1,p2:integer;                                                   {variable for school info}
    s,s1,s2:integer;                                                  {variable for student info and
ans}
    a,a1:integer;                                                      {variable for answer key}
    c:integer;                                                           {variable for counting}
    num_sch,num_stu:integer;                                           {Number of schools and
students}
    Sa,Ori:string;                                                    {Vaviable of cheking answer}
    marks:array [1..max_part] of integer;                  {Marks count for stuent}
    marks_s:array [1..max_part] of integer;               {Marks for sorting}
    place:array [1..max_part] of integer;                 {Order of sorting}
    Q_marks:array [1..max_question] of integer;             {Marks for calculating
percentage}
    Q_percen:array [1..max_question] of real;            {Percentage correct of each
question}
    i,j,x,y,z:integer;                                              {Array variable}
    freq:array [1..max_part] of integer;                          {For calculate number of
participants for each school}
    sch_marks:array [1..max_sch] of integer;              {For calculate total marks of each
school}
    ave_marks:array [1..max_sch] of real;                 {For calculate average marks of
each school}
    chart_sch_marks:array [1..max_sch] of integer;    {Value of average marks in chart}

Procedure Auto_read; {read participants' Automatically}

Var
    sch_info:text;    {info of schools}
    stu_info:text;    {info of participants and their ans}
    ans_key:text;      {answer key of mc}
    filename1,filename2,filename3:string; {filename of text file}

   Begin
       writeln('Please enter the filename which contain following information.');
       write('Information of schools:');
       filename1:='sch_info.txt';
       assign(sch_info,filename1);
       reset(sch_info);
       p:=0;
       while not eof(sch_info) do
               begin
                     p:=p+1;
                     readln(sch_info,sch_code[p],sch_name[p]);
               end;
       write('Answer of participants:');
       filename2:='stu_ans.txt';
       assign(stu_info,filename2);
       reset(stu_info);
       s:=0;
       while not eof(stu_info) do
               begin
                     s:=s+1;
                     readln(stu_info,sch_code_stu[s],stu_code[s],stu_name[s],stu_ans[s]);
               end;
        write('Answer Key file:');
        filename3:='anskey.txt';
        assign(ans_key,filename3);
        reset(ans_key);
        a:=0;
        while not eof(ans_key) do
                begin
                       a:=a+1;
                       readln(ans_key,ori_ans);
                end;

   End;
```

```
Procedure Manual_read ; {read participants' info Manually}

Var
    sch_info:text;    {info of schools}
    stu_info:text;    {info of participants and their ans}
    ans_key:text;      {answer key of mc}
    filename1,filename2,filename3:string; {filename of text file}

   Begin
       clrscr;
       writeln('Please enter the filename which contain following information.');
       write('Information of schools:');
       readln(filename1);
       assign(sch_info,filename1);
       reset(sch_info);
       p:=0;
       while not eof(sch_info) do
               begin
                     p:=p+1;
                     readln(sch_info,sch_code[p],sch_name[p]);
               end;
       write('Answer of participants:');
       readln(filename2);
       assign(stu_info,filename2);
       reset(stu_info);
       s:=0;
       while not eof(stu_info) do
               begin
                     s:=s+1;
                     readln(stu_info,sch_code_stu[s],stu_code[s],stu_name[s],stu_ans[s]);
               end;
        write('Answer Key file:');
        readln(filename3);
        assign(ans_key,filename3);
        reset(ans_key);
        a:=0;
        while not eof(ans_key) do
                begin
                      a:=a+1;
                      readln(ans_key,ori_ans);
                end;

   End;


Procedure Met_read;    {Choose method for read in data}
```

```
Var
   met:char;
   Begin
      writeln('Choose a method to input data:');
      writeln('A. Auto-Input                             (Choose this if filename is default.)');
      writeln('M.  Manual-Input                                   (Choose   this   if   filename   is
modified.)');
      readln(met);
      case met of
      'a','A':Auto_read;
      'm','M':Manual_read;
      else
         begin
            writeln;
            writeln('*************************************');
            writeln('***********Invalid output************');
            writeln('******Auto return in one second******');
            writeln('*************************************');
            delay(1000);
            clrscr;
            met_read;
         end;
      end
   end;


Procedure List_sch; {list out school information}

   Begin
        writeln('School Code             School Name');
        for p1:=1 to p do
                writeln(sch_code[p1]:6,sch_name[p1]:23);

   End;


Procedure List_stu; {list out school information and answer}

   Begin
        writeln('School  Code             School  Name           Student  Code           Student
Name');
        for s1:=1 to s do
              begin
                s2:=sch_code_stu[s1];

writeln(sch_code_stu[s1]:6,sch_name[s2]:23,stu_code[s1]:12,stu_name[s1]:27)
```

```
         end;

  End;


Procedure Count_sch_part; {Count the number of participants and schools}

  Begin
      for c := 1 to max_part do
      begin
       if (sch_code_stu[c]<>sch_code_stu[c-1]) and (sch_code_stu[c]>0)
         then num_sch:=num_sch+1;
       if sch_code_stu[c]>0
         then num_stu:=num_stu+1;
      end;     writeln('Total number of school(s):',num_sch,'   Total number of
participant(s):',num_stu);
  End;

Procedure Show_sch_part;   {List the total number of participants and school'}

  Begin
    writeln('Total number of school(s):',num_sch,'   Total number of
participant(s):',num_stu);
  End;

Procedure answer_checker;   {check students answer}

  Begin
    for x:=1 to s do
      begin
        marks[x]:=0;
        for y:=1 to max_question do
          begin
            Ori:=copy(ori_ans,y,1);
            Sa:=copy(stu_ans[x],y+1,1);
            if ori=Sa
              then marks[x]:=marks[x]+1;
          end;
       end;
  end;

Procedure percentage_checker;   {check students answer}

  Begin
    for y:=1 to max_question do
      begin
```

```
        for z:=1 to s do
          begin
            Ori:=copy(ori_ans,y,1);
            Sa:=copy(stu_ans[z],y+1,1);
            if ori=Sa
              then Q_marks[y]:=Q_marks[y]+1;
          end;
       end;
    for y:=1 to max_question do
      begin
        Q_percen[y]:=Q_marks[y]/s*100;
      end;
  end;

Procedure show_ans;      {To list out the answer key and students answer}

   Begin
     writeln('Answer key');
     writeln(ori_ans);
     writeln('student number    Number of correct                  Answer of student');
     for x:=1 to s do
       begin
         writeln(x:7,marks[x]:13,stu_ans[x]:max_question+5);
       end;
   End;

Procedure show_percen;    {To list the coorect percentage of each question}

  Begin
    writeln('Question No.    No.(s) of Correct    Correct percentage');
    for y:=1 to max_question do
      begin
        writeln(y:11,Q_marks[y]:14,Q_percen[y]:19:3);
      end;
  End;

Procedure Sort_stu_marks; {For sorting}
var temp_stumark:integer;
    temp_stucode:string;
  Begin
    for x:=1 to s do
      begin
       marks_s[x]:=marks[x];
       stu_code_s[x]:=stu_code[x];
      end;
        for z:=1 to s-1 do
```

```
               for    x:=1 to s-z do
                  if marks_s[x]<marks_s[x+1] then
                     begin
                        temp_stumark:=marks_s[x];
                        marks_s[x]:=marks_s[x+1];
                        marks_s[x+1]:=temp_stumark;
                        temp_stucode:=stu_code_s[x];
                        stu_code_s[x]:=stu_code_s[x+1];
                        stu_code_s[x+1]:=temp_stucode;
                     end;
   End;

Procedure Total_stu_sch;      {Calculate the total number of students form each school}

   Begin
      for p1:=1 to p do
         for s1:=1 to s do
            begin
               if sch_code_stu[s1]=sch_code[p1] then
                  freq[p1]:=freq[p1]+1;
            end;
   End;

Procedure Average_marks_sch;        {Calculate average marks of school}
   Begin
      for p1:=1 to p do
         for s1:=1 to s do
            begin
               if sch_code_stu[s1]=sch_code[p1] then
                    sch_marks[p1]:=sch_marks[p1]+marks[s1];
            end;
         for p1:=1 to p do
            begin
               if freq[p1]<>0 then
               ave_marks[p1]:=sch_marks[p1]/freq[p1];
            end;
   End;

Procedure Total_stu_sch_list;       {To list out total number of participants from each school}

   Begin
      writeln('School code    School name    Number of participants    Average marks');
      for p1:=1 to p do
         begin
            writeln(sch_code[p1]:6,sch_name[p1]:18,freq[p1]:10,ave_marks[p1]:25:3);
         end;
   End;


Procedure Pre_chart;              {Calculate the value for chart}

   Begin
      for p1:=1 to p do
         begin
            if freq[p1]<>0 then
               chart_sch_marks[p1]:=sch_marks[p1] div freq[p1];
         end;
   End;

Procedure Chart_sch;              {Chart for school average marks}

   Begin
      writeln('The following chart show the average marks of each school');
      writeln('School code    Marks');
      for p1:=1 to p do
         begin
            write(sch_code[p1]);
            write('                    ');
            for p2:=1 to chart_sch_marks[p1] do
               begin
                  write('*');
               end;
            write('    ',chart_sch_marks[p1]);
            writeln;
         end;
   End;




Procedure List_place; {List the order of student according to marks}

   Begin
      writeln('First    Student number');
      writeln('1':4,stu_code_s[1]:11);
      writeln('Second    Student number');
      writeln('2':4,stu_code_s[2]:11);
      writeln('First    Student number');
      writeln('3':4,stu_code_s[3]:11);
      writeln('Place    Student number');
      for x:=4 to s do
      writeln(x:4,stu_code_s[x]:11);
  End;

Procedure Output_report;     {To save the report in a text file}
var
    name:string;
    report:text;
  Begin
    writeln('Please give a name to the output file:');
    readln(name);
    name:=name+'.txt';
    assign(report,name);
    rewrite(report);
      begin
        writeln(report,'1.The total number of schools & students');

writeln(report,'*************************************************************
****');
        writeln(report,'Total number of school(s):',num_sch,'    Total number of
participant(s):',num_stu);
        writeln(report);

        writeln(report,'2.The information of school');

writeln(report,'*************************************************************
****');
        writeln(report,'School Code              School Name');
          for p1:=1 to p do
            writeln(report,sch_code[p1]:6,sch_name[p1]:23);
        writeln(report);

        writeln(report,'3.The information of participants');

writeln(report,'*************************************************************
****');
        writeln(report,'School Code                School Name              Student Code
Student Name');
          for s1:=1 to s do
           begin
            s2:=sch_code_stu[s1];

writeln(report,sch_code_stu[s1]:6,sch_name[s2]:23,stu_code[s1]:12,stu_name[s1]:27)
           end;
        writeln(report);

        writeln(report,'4.Total number of students from each school');
writeln(report,'****************************************************************
****');
        writeln(report,'School code    School name    Number of participants    Average
marks');
          for p1:=1 to p do
            begin

writeln(report,sch_code[p1]:6,sch_name[p1]:18,freq[p1]:10,ave_marks[p1]:25:3);
            end;
        writeln(report);

        writeln(report,'5.Answer key & Student result');

writeln(report,'****************************************************************
****');
        writeln(report,'Answer key');
        writeln(report,ori_ans);
        writeln(report,'student number    Number of correct    Answer of students');
          for x:=1 to s do
            begin
              writeln(report,x:7,marks[x]:17,stu_ans[x]:Max_question+11);
            end;
        writeln(report);

        writeln(report,'6.Correct answer percentage');

writeln(report,'****************************************************************
****');
        writeln(report,'Question No.    No.(s) of Correct    Correct percentage');
          for y:=1 to max_question do
            begin
              writeln(report,y:11,Q_marks[y]:14,Q_percen[y]:19:3);
            end;
        writeln(report);

        writeln(report,'7.Show place of student');

writeln(report,'****************************************************************
****');
        writeln(report,'First    Student number');
        writeln(report,'1':4,stu_code_s[1]:11);
        writeln(report,'Second    Student number');
        writeln(report,'2':4,stu_code_s[2]:11);
        writeln(report,'First    Student number');
        writeln(report,'3':4,stu_code_s[3]:11);
```

```
            writeln(report,'Place    Student number');
               for x:=4 to s do
                  writeln(report,x:4,stu_code_s[x]:11);
            writeln(report);

            writeln(report,'8.Chart showing school average marks');

writeln(report,'****************************************************************
****');
            writeln(report,'The following chart show the average marks of each school');
            writeln(report,'School code    Marks');
               for p1:=1 to p do
                  begin
                     write(report,sch_code[p1]);
                     write(report,'                  ');
                     for p2:=1 to chart_sch_marks[p1] do
                        begin
                           write(report,'*');
                        end;
                     write(report,'    ',chart_sch_marks[p1]);
                     writeln(report);
                  end;
            writeln(report);


         end;
      close(report);
   End;


Procedure Menu;    {Menu for choosing data to output}
Var
   Q:char;
   Deci:char;
   Begin
      clrscr;
      writeln('Here are the list of number you can choose from.');
      writeln('Please choose it wisely.');
      writeln('1. List all school code and name');
      writeln('2. List all student information');
      writeln('3. Total number of schools and participants');
      writeln('4. Total number of students from each school');
      writeln('5. Show answer key and student result');
      writeln('6. Show correct answer percentage');
      writeln('7. Show place of student');
      writeln('8. Show chart of school average marks');
```

```
    writeln('9. Save an external analysis report');
    writeln('0. EXIT');
    write('Please choose a number from above:');
    readln(Q);
    case Q of
    '1':begin
        writeln;

writeln('************************************************************');
        writeln;
        List_sch;
        writeln('Press <enter> to continue');
        readln;
        Menu;
      end;
    '2':begin
        writeln;

writeln('************************************************************');
        writeln;
        List_stu;
        writeln('Press <enter> to continue');
        readln;
        Menu;
      end;
    '3':begin
        writeln;

writeln('************************************************************');
        writeln;
        Show_sch_part;
        writeln('Press <enter> to continue');
        readln;
        Menu;
      end;
    '4':begin
        writeln;

writeln('************************************************************');
        writeln;
        Total_stu_sch_list;
        writeln('Press <enter> to continue');
        readln;
        Menu;
      end;
    '5':begin
```

```
                writeln;

writeln('**********************************************************************');
                writeln;
                Show_ans;
                writeln('Press <enter> to continue');
                readln;
                Menu;
            end;
        '6':begin
                writeln;

writeln('**********************************************************************');
                writeln;
                Show_percen;
                writeln('Press <enter> to continue');
                readln;
                Menu;
            end;
        '7':begin
                writeln;

writeln('**********************************************************************');
                writeln;
                List_place;
                writeln('Press <enter> to continue');
                readln;
                Menu;
            end;
        '8':begin
                writeln;

writeln('**********************************************************************');
                writeln;
                Chart_sch;
                writeln('Press <enter> to continue');
                readln;
                Menu;
            end;
        '9':begin
                writeln;

writeln('**********************************************************************');
                writeln;
                Output_report;
                writeln('Press <enter> to continue');
        readln;
        Menu;
      end;
    '0':begin
       writeln('Are you sure that you want close the program?');
       writeln('Please enter Y/N...');
       readln(Deci);
         case Deci of
         'Y','y':begin
                 textcolor(12);
                 writeln('**********PROGRAM TERMINATED**********');
                 delay(1000);
                 writeln('**********CONTROL OFFLINE*************');
                 delay(1000);
                 end;
         'N','n':Menu;
                 else
                 writeln('Invalid choice');
                 writeln('Auto return to Menu');
                 delay(1200);
                 Menu;
         end
       end;
    else
      writeln('I do not realise there is choice ''',Q,'''');
      writeln('Please enter an integer again');
      writeln('*****************************************************');
      writeln('Auto return to Menu');
      delay(1200);
      Menu;
    end
  End;

Procedure Ran_num;       {On screen random number wallpaper}
var
  Num1,Num2:integer;
  Begin
    clrscr;
    randomize;
    for Num1:=1 to 60 do
      begin
        randomize;
        for Num2:=1 to 78 do
          write(random(2));
        delay(Num1 div 2);
        writeln;
      end;
    delay(200);
    clrscr;
    randomize;
    writeln('Loading        *****000%*****');
    delay(500);
    writeln('Loading        *****01',random(10),'%*****');
    delay(500);
    writeln('Loading        *****03',random(10),'%*****');
    delay(500);
    writeln('Loading        *****06',random(10),'%*****');
    delay(500);
    writeln('Loading        *****07',random(10),'%*****');
    delay(500);
    writeln('Loading        *****100%*****');
    delay(500);
    writeln('**********Loading complete**********');
    delay(1000);
  End;



Begin {main program}

      textbackground(9);
      textcolor(14);
      clrscr;
      Met_read;
      Answer_checker;
      Percentage_checker;
      Sort_stu_marks;
      Total_stu_sch;
      Average_marks_sch;
      Count_sch_part;
      Pre_chart;
      Ran_num;
      Menu;
End.
```

Appendix 2 - Working schedule

Date	Items to be submitted	Tasks to be completed
18-Dec-2008		Briefing Session
18-Jan-2009	1st Report – Hard & Soft copy	Chapter 1 to Chapter 3 (Objective, Analysis & Design)
5-Feb-2009	Program & Data files	Chapter 4 (Implementation)
8-Feb-2009	2nd Report – Hard copy & Soft copy	Chapter 5 Testing and Evaluation
11-Feb-2009	Draft of complete report & program	Chapter 1 to Chapter 7
15-Feb-2009	Coursework & Final Report	See the requirements in the question paper (Refer to the Guideline)