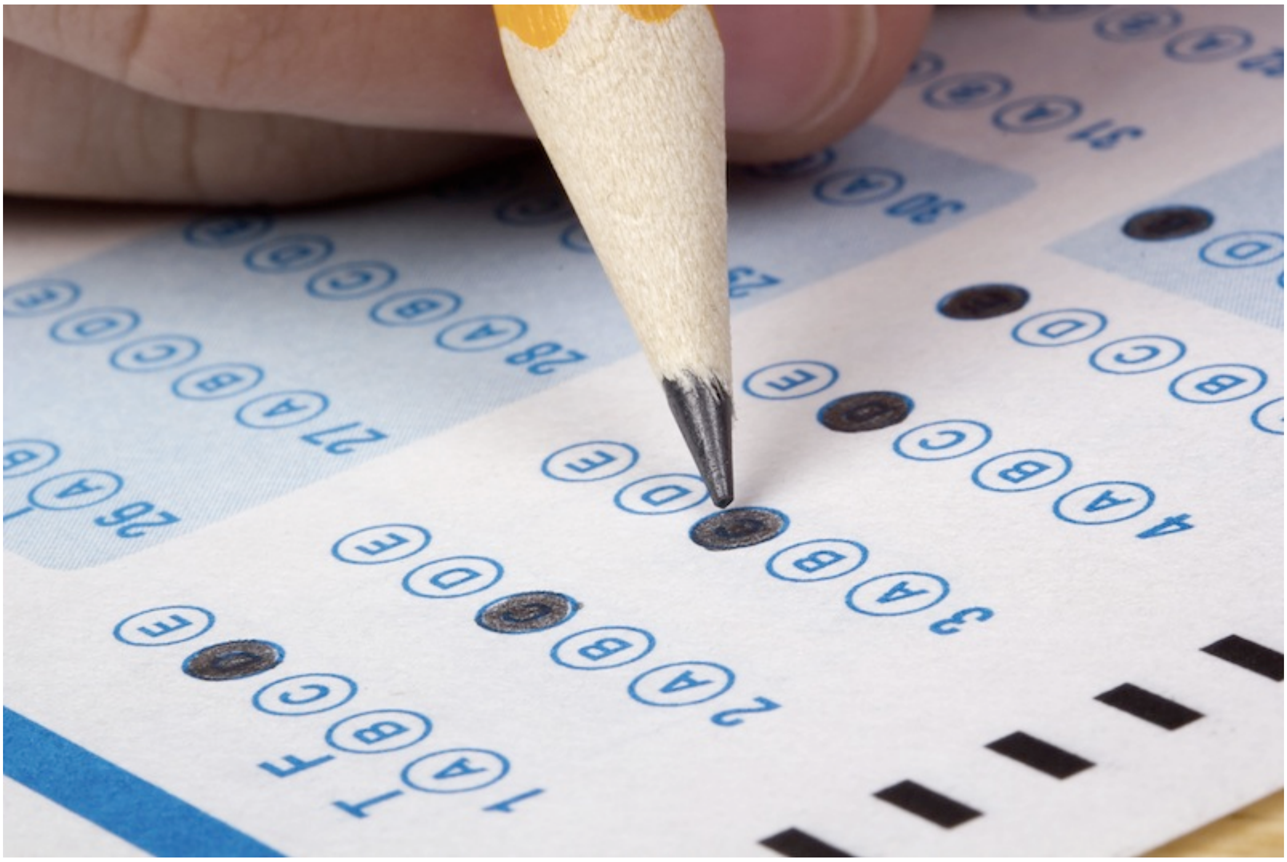
Multiple Choice Analysis System

2015 HKDSE
Information & Communication Technology
School Based Assessment
Elective D (Software Development)

S.K.H. Chan Young Secondary School

6C Chan Wing Hang
Date: 1/10/2014



Introduction

Multiple choice questions ask people to choose the answer from a list. It mostly used in educational testing. I am going to develop a computer program to read all raw plain text data and produce a detailed analysis report on the competition. I need to write a program that the answer sheets read and converted into a text file using a multiple choice analysis system.

The multiple choice analysis system can provide different functions, including:

1. Question analysis
2. Searching Student
3. Showing the result
4. Exporting Report

Competition regulations

Individual awards

1. A student who gets the highest mark can get the champion, the second one can get the 1st runner up and the third one will get 2nd runner up.
2. If more than one student gets the same mark, they will get the same rank and the rank of the following students will be skipped. (Three students get champion, the fourth one will get 3rd runner up)

School awards

1. A school which can get the highest mean can get the champion, the second one can get the 1st runner up and the third one will get the 2nd runner up.
2. If more than one school gets the same mean, they will get the same rank and the rank of the following school will be skipped. (Three schools get champion, the fourth one will get the 3rd runner up)

Design

Text files' format

```
1.txt
Name
School
Answer
(Example: ABCDABCDAB)
```

Fig. 1

Each file(Fig. 1) represents each MC sheet. The file name represents the student ID. The file '0.txt' the model answer, user can update the model answer to different questions. The first line and second line represent the student's name and school. The third line is student's answers which are written in a single line. The structure is clear that the student can modify the content easily.

Main Functions

Question Analysis

"Question Analysis" shows the correct answer for each question and allows user to check the number of students get the answer correctly, and shows the percentage of overall students who answer the right answer. It helps teachers to find the hardnesses of questions and students' weaknesses. When the system starts, it processes all data automatically

Search Students & Schools

This function allows users to enter the student's name and the program will find out the information about individual student such as name, school, rank and answers. Thanks for array structure, the target can be easily found as the data are stored tidily. It is a very convenient function.

Performance

By using the "Performance", the system shows all the marks, ranks and marks of each student. Moreover, the function also shows the number of students, the mark and rank for each school.

Prize

In the competition, the students and schools who get good performance, they will get prize for encouragement. The system will analysis which students and school get good result and show the prize they can get. Other details can be found in "Performance" and "Search Students".

Export Result

Result can be export in text file as the result can be printed and backup all the information. There are three parts, the first will show the number of participants, participating schools and participant(s) from each participating school. The second part will show winners of individual awards and school awards. The third part will show the result of question analysis. It is very clean to show all the information. The program will not store the data because the system is reusable, users no need to clear the data for next time.

Algorithm Design

Sorting

After the marking and calculation, marks will be generated. To calculate student's rank, sorting is very important. To do this, I will try to copy students' marks to an array. The program will use "Insertion Sort" to sort the whole set of data in ascending order. Then it will compare the array with students' marks. The program will give ranks for students and schools.

Searching

To check student's answer and find student's information, searching is a very important function. In this program, data are not very large as it is designed for small amount of students, 'binary search' is used as it is easy to write and process time is not very long. After the program have inputted all data, it will search modal answer for each answer, if student's answer is not the same as the modal answer, mark will not be added.

Data Structure

Text files

A student's answers and basic information stores in one text file. The text files will read by the system can input the data process. In reality, people use physical paper to scan the sheets. Because of technical limiting, I use a text file to replace one MC answer sheet. File marge is not suitable in this system as it is not logical to marge two MC answer sheet into one file. Also the result can be output in text file because the result can be printed and store in hard disk for backup. Text file is a good file type because of machine independent, files can be read by all computers ever through the computers haven't install the multiple choice analysis system.

Records and Array

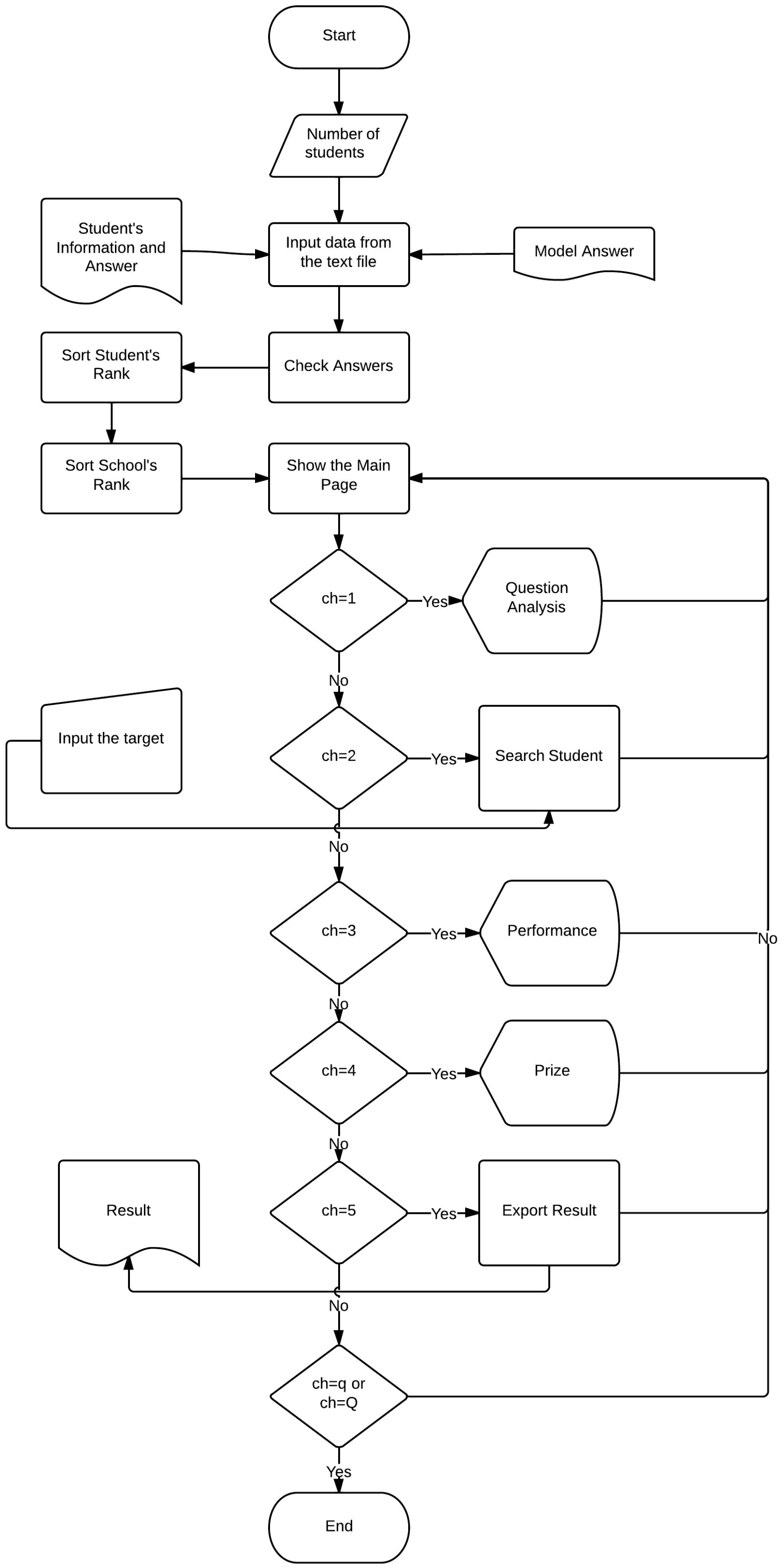
'Records' is very important for multiple choice analysis system as there are many fields. It include student's name, school, mark and answers. It is very convenient to arrange an array of records to store student's name, school and answer. Thanks for records, the system can find student's and school's information easily. The reason why the program haven't use not two-dimensional array is because the system only process one quiz, it doesn't used to compare with two result. One-dimensional array is sufficient for writing the program.

String

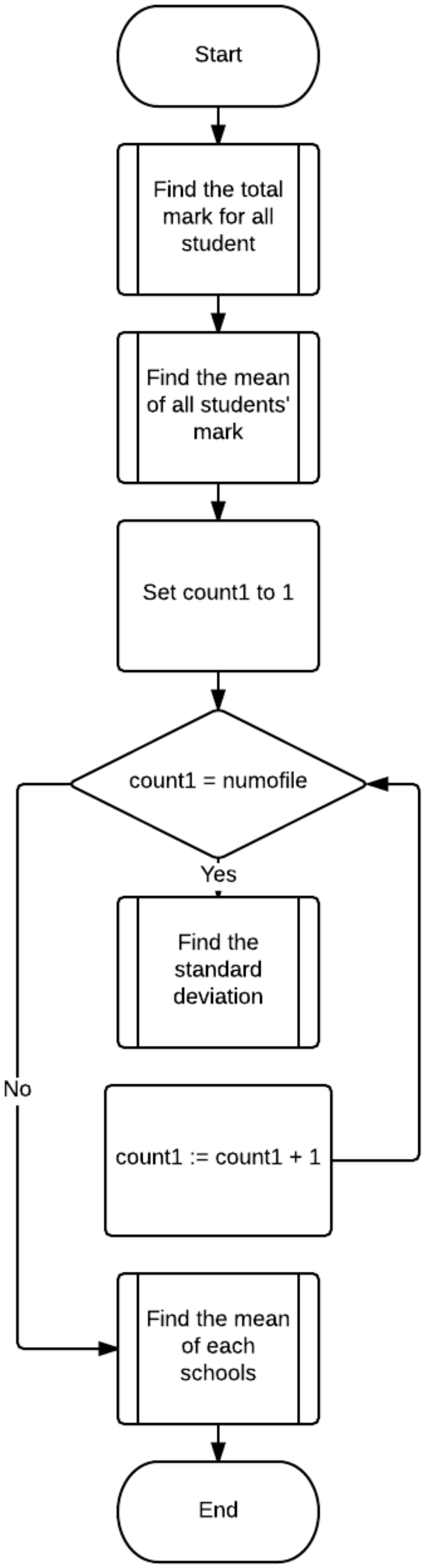
In 'Records', all data are stored in integer type and string type. The most important part to use string is store students' answers. For example the system need to compare students' answers with model answer. It is well organized students' answer that computer can process the data easily. Programmers can `student[1].answer[3]`, so that computer can find out the variable directly.

Flowchart

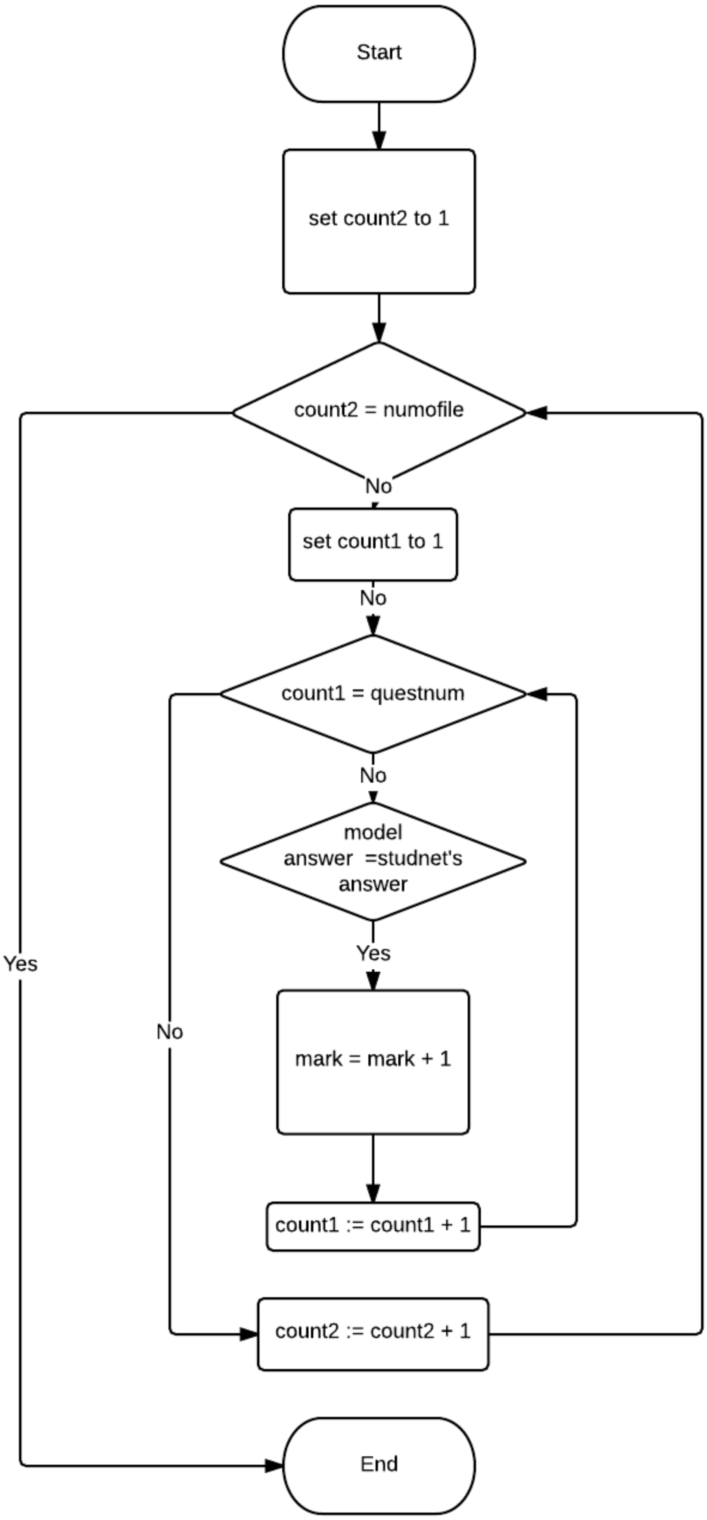
The whole system



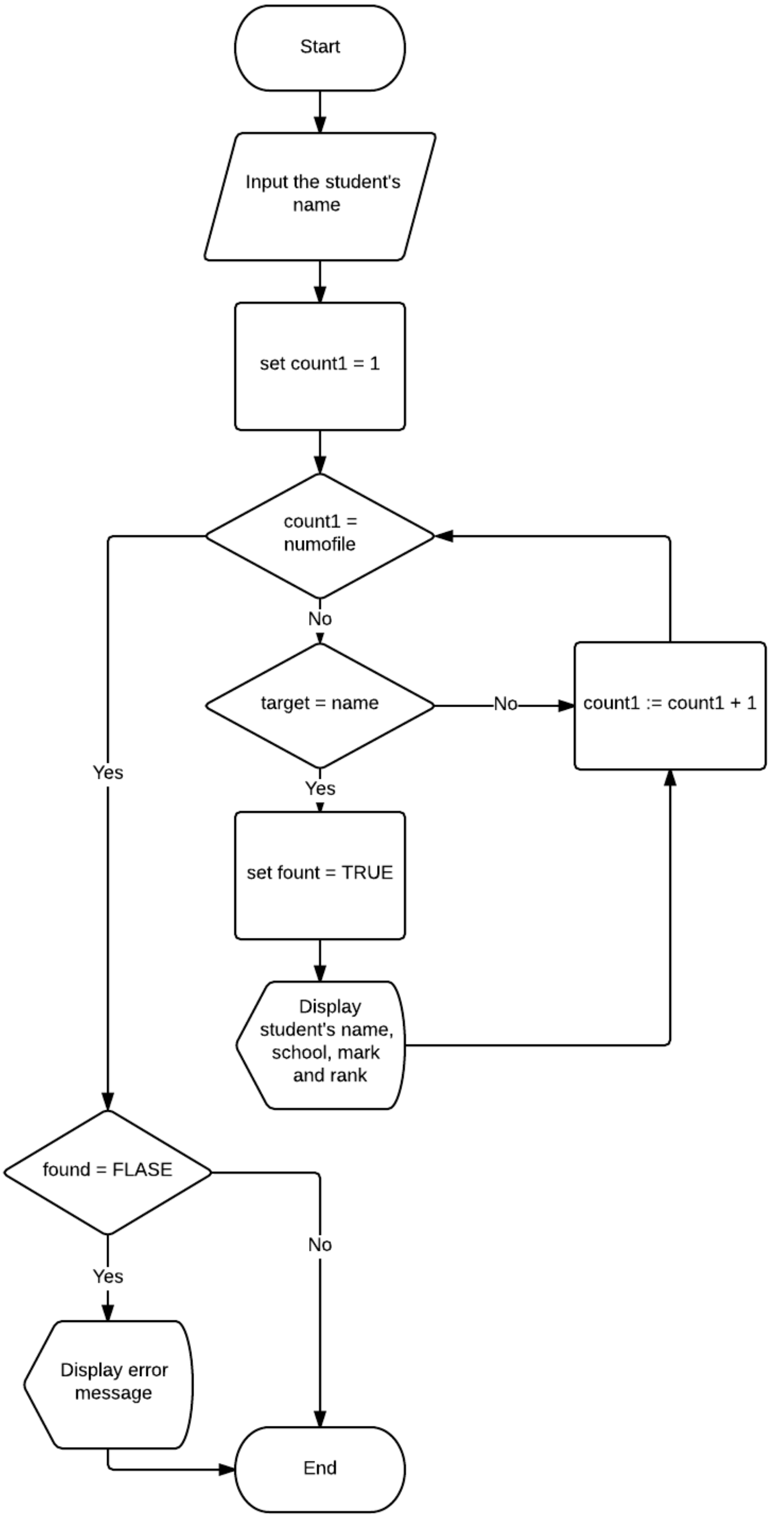
Statistics


Start
Find the total mark for all student
Find the mean of all students' mark
Set count1 to 1
count1 = numofile
Yes
Find the standard deviation
No
count1 := count1 + 1
Find the mean of each schools
End

Check Answer



Search a Student



Program Code

——— Internal Procedure

——— UI Procedure

```
program readfile;
uses crt, sysutils, dos;
type
      StudentType = Record
                  ID : string;
                  Name : string;
                  School : string;
                  Mark : integer;
                  Answer : string;
                  Rank : integer;
         StandScore : real;
                  end;
      SchoolType = Record
                  Name : string;
         StudentID : array[0..30] of string;
                  Freq : integer;
                  Mark : integer;
                  Rank : integer;
         Mean : real;
                  end;
```

```
var
      Df : text;
      Student : array[0..30] of StudentType;
      School : array[1..10] of SchoolType;
      Answerfreq : array[1..50] of integer;
      numofile : integer;
      questnum : integer;
      count1, count2, count3 : integer;
      schoolnum : integer;
      mean : real;
      standev : real;
      ch : char;
      Year, Month, Day, WDay : Word;
```

Global Variable List

```
procedure inputans(filename : string);
var
      iNum : integer;
      code : integer;
begin
      {Assign files and input students' name, school and answers}
      assign(Df, 'E:\'+filename+'.txt');
      {$I-}
      reset(Df);
      {$I+}
      if (IOResult <> 0) then
      begin
              writeln('Mmmm... Some files are missing. Exiting...');
              readln;
      end else
      begin
        val(filename, iNum, code);
        with student[iNum] do
        repeat
             ID := filename;
             readln(Df, Name);
             readln(Df, School);
             readln(Df, answer);
        until eof(Df);
      end;
      close(Df);
end;
```

"IOResult", "{$I-}", and "{$I+}" is used avoid run-time error when the file can't be found.

```
procedure filenum;
var
      files : string;
begin
        clrscr;
        {Input the Num of file and change integer into string for filename}
        writeln('Welcome! Thank you for using Multiple-Choice Analysis 
System!');
        write('Please input the number of students:   ');
        readln(numofile);
        write(‘Please input the number of question:   ');
        readln(questnum);
        while (numofile = 0) or (questnum = 0) do
        begin
                   writeln('Input error!');
                   write('Input the number of students:   ');
                   readln(numofile);
          write('Please input the number of question:   ');
          readln(questnum);
        end;

        for count1 :=0 to numofile do
        begin
             str(count1, files);
             inputans(files);
        end;
end;
```

“while (numofile=0) or (quesrnum=0)” is used to avoid the program crash when they input ‘0’.

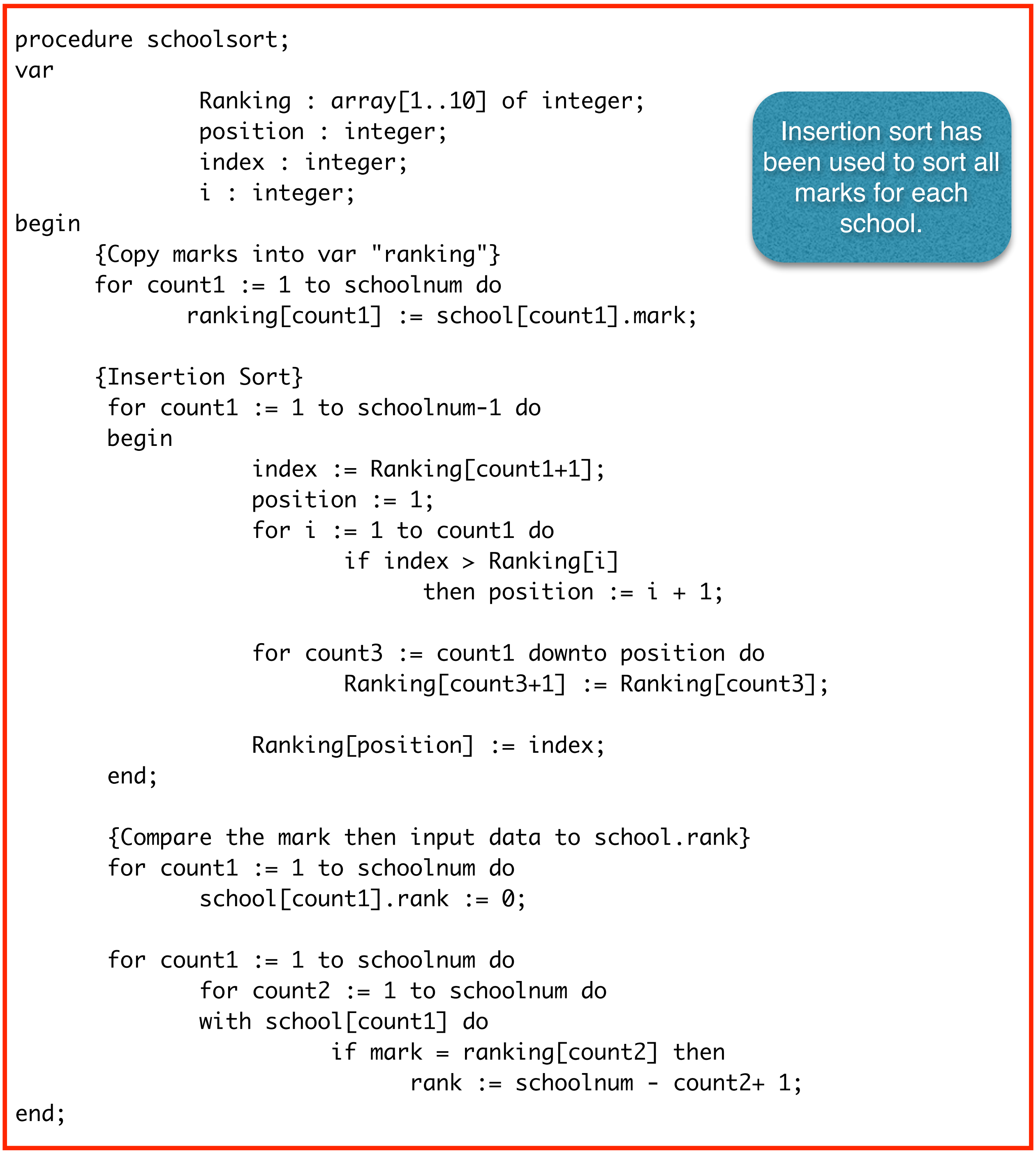
```
procedure checkans;
begin
      for count1 := 1 to questnum do
      begin
              answerfreq[count1] := 0;
              student[count1].mark := 0;
      end;

      {Check students' answers and score}
      for count2 := 1 to numofile do
      with student[count2] do
      begin
            for count1 := 1 to questnum do
                  if answer[count1] = student[0].answer[count1] then
                  begin
                          mark := mark + 1;
                          answerfreq[count1] := answerfreq[count1] + 1;
                  end;
      end;
end;
```

```
procedure studsort;
var
              Ranking : array[1..10] of integer;
              position : integer;
              index : integer;
              i : integer;
begin
      {Copy marks into var "ranking"}
      for count1 := 1 to numofile do
             ranking[count1] := student[count1].mark;

      {Insertion Sort}
       for count1 := 1 to numofile-1 do
       begin
                  index := Ranking[count1+1];
                  position := 1;
                  for i := 1 to count1 do
                         if index > Ranking[i]
                               then position := i + 1;

                  for count3 := count1 downto position do
                         Ranking[count3+1] := Ranking[count3];

                  Ranking[position] := index;
       end;

       {Compare the mark then input data to student.rank}
       for count1 := 1 to numofile do
              student[count1].rank := 0;

       for count1 := 1 to numofile do
              for count2 := 1 to numofile do
              with student[count1] do
                        if mark = ranking[count2] then
                              rank := numofile - count2+ 1;
end;
```

Insertion sort has been used to sort all marks for each student.

This procedure is used to count the number of school, the number of students in the school and marks school can get.

```
procedure schoolfreq;
var
       Found : Boolean;
begin
       {initialization}
       for count1 := 1 to numofile do
       begin
              School[count1].Name := '';
              School[count1].Freq := 0;
              School[count1].Mark := 0;
       end;
       schoolnum := 1;
       School[1].Name := Student[1].School;
   School[1].StudentID[1] := Student[1].ID;

       for count1 := 1 to numofile do
       begin
              Found := FALSE; {initialization for FOUND}
              for count2 := 1 to schoolnum do
                      if Student[count1].School = School[count2].Name then
                             begin
                                         School[count2].Freq := School[count2].Freq +
1;
                                         School[count2].Mark := School[count2].Mark +
Student[count1].Mark;
                   {To link the relationship between school's data and student's data}
                   School[count2].StudentID[School[count2].Freq] :=
Student[count1].ID;
                                         Found := TRUE;
                             end else
                                         if (count2 = schoolnum) and (Found = FALSE)
then
                                         begin
                                                     schoolnum := schoolnum + 1;
                                                     School[schoolnum].Name :=
Student[count1].School;
                                                     School[schoolnum].Mark :=
Student[count1].Mark;
                        {To link the relationship between school's data and student's
data}
                        School[schoolnum].StudentID[1] := Student[count1].ID;
                                                     School[schoolnum].Freq := 1;
                                         end;
       end;
end;
```



```
procedure schoolsort;
var
              Ranking : array[1..10] of integer;
              position : integer;
              index : integer;
              i : integer;
begin
      {Copy marks into var "ranking"}
      for count1 := 1 to schoolnum do
             ranking[count1] := school[count1].mark;

      {Insertion Sort}
       for count1 := 1 to schoolnum-1 do
       begin
                  index := Ranking[count1+1];
                  position := 1;
                  for i := 1 to count1 do
                         if index > Ranking[i]
                               then position := i + 1;

                  for count3 := count1 downto position do
                         Ranking[count3+1] := Ranking[count3];

                  Ranking[position] := index;
       end;

       {Compare the mark then input data to school.rank}
       for count1 := 1 to schoolnum do
              school[count1].rank := 0;

       for count1 := 1 to schoolnum do
              for count2 := 1 to schoolnum do
              with school[count1] do
                        if mark = ranking[count2] then
                              rank := schoolnum - count2+ 1;
end;
```

Insertion sort has been used to sort all marks for each school.

```
procedure statistics;
begin
     mean := 0;
     standev := 0;
     {Find Mean}
     for count1 := 1 to numofile do
         mean := mean + student[count1].mark;
     mean := mean / numofile;
     {Find standard deviation}
     for count1 := 1 to numofile do
         standev := sqr(student[count1].mark - mean) + standev;
     standev := sqrt(standev / numofile);
     for count1 := 1 to schoolnum do
     with school[count1] do
         mean := mark / freq;
end;
```

```
procedure start;
begin
        filenum;
        checkans;
        studsort;
        schoolfreq;
        schoolsort;
        statistics;
end;
```

This procedure is used to integrated all procedure which run before the main page is shown.

```
{==============The end of Background Procedures=================}
{===============The start of Visible Procedures=================}
```

```
procedure style;
begin
        TextBackground(Blue);
        TextColor(Yellow);
        clrscr;
end;
```

‘Style’ is used to color the background and texts.

```
procedure qanalysis;
var percent : integer;
    item : integer;
begin
          style;
     gotoxy(33,5);writeln('Question Analysis');
          gotoxy(23,7);writeln('Question', 'Freq':6, 'Answer': 10, 'Percentage': 13);
     item := 6;
     count2 := 0;
     repeat
          for count1 := (item-5) to item do
              begin
                           percent := answerfreq[count1] * 100 div 10;
           count2 := count2 + 1;
                           gotoxy(26,7+count2);
           writeln(count1:5, answerfreq[count1]:6 , student[0].answer[count1]: 10, percent: 12,'%');
         end;
         gotoxy(26,9+count2);
         count2 := 0;
         item := item + 6;
         writeln('Press enter to continue.');
         ch := readkey;
     until (item >= questnum+6);
end;
```

```
procedure searchstu;
var target : string;
    found : Boolean;
begin
     clrscr;
     found := FALSE;
         gotoxy(24,7);writeln('Please enter the participants'' full name.');
     gotoxy(24,8);readln(target);
     gotoxy(24,10);
     for count1 := 1 to numofile do
         with student[count1] do
         if target = name then
     begin
               found := TRUE;
            gotoxy(24,10);writeln('Name: ', name);
            gotoxy(24,11);writeln('School: ', school);
            gotoxy(24,12);writeln('Mark: ', mark);
            gotoxy(24,13);writeln('Rank: ',rank);
     end;
         if found = FALSE then writeln('Oops! No result can be shown.');
     gotoxy(24,15);
     writeln('Press enter to continue.');
     ch := readkey;
end;
```

In this procedure, gotoxy(x,y) is used as typesetting.

```
procedure searchsch;
var target : string;
    found : Boolean;
begin
    clrscr;
        gotoxy(24,7);writeln('Please enter the schools'' full name.');
    gotoxy(24,8);readln(target);
    gotoxy(24,10);
    found := FALSE;
    for count1 := 1 to schoolnum do
    with school[count1] do
    if target = name then
        begin
          found := TRUE;
      gotoxy(24,10);writeln('Name: ', name);
             gotoxy(24,11);writeln('No. of participants: ', freq);
             gotoxy(24,12);writeln('Mark: ', mark);
             gotoxy(24,13);writeln('Rank: ',rank);

      {Shows all participants inside the school}
      gotoxy(24,15);writeln('Student');
      gotoxy(24,16);writeln('ID':2, 'Name':20, 'Mark':6, 'Rank':6);
      for count2 := 1 to freq do
          begin
             gotoxy(24,17+count2);
             write(school[count1].studentID[count2]:2);
             with student[count2] do
             writeln(Name:20, Mark:6, Rank:6);
          end;

       end;
   if found = FALSE then writeln('Oops! No result can be shown.');
   ch := readkey;
end;
```

```
procedure search;
begin
     style;
     gotoxy(34,5);writeln('Searching');
     gotoxy(28,7);writeln('1. Participant');
     gotoxy(28,8);writeln('2. School');
     ch := readkey;
     if ch = '1' then searchstu else
        if ch = '2' then searchsch;
end;
```

Press ‘2’ in mainpage, the system will operate this procedure to search student.

```
procedure performance;
var item : integer;
begin
    style;
    gotoxy(37,5); writeln('Performance');
    gotoxy(30,7); writeln('Participants'' Performance');
    gotoxy(15,8); writeln('No. of participants: ', numofile);
    gotoxy(15,9); writeln('No. of participating schools: ',schoolnum);
    gotoxy(15,10);writeln('Mean: ',mean :0:1);
    gotoxy(15,11);writeln('Standard deviation: ', standev :0:1);

        {Output result of student}
        gotoxy(15,13); writeln('Name':15, 'Mark':6, 'School':15, 'Rank':6);
    item := 7;
    count2 := 0;
    repeat
      for count1 := (item-6) to item do
      with student[count1] do
      begin
         count2 := count2 + 1;
             gotoxy(15,13 + count2);
         write(Name :15, Mark :6, School :15, Rank :6);
      end;
      item := item + 7;
      gotoxy(15,14 + count2);
      count2 := 0;
      writeln('Press enter to continue.');
      ch := readkey;
    until item >= numofile+7;

         {Output result of school}
     clrscr;
     gotoxy(32,5); writeln('School''s Performance');
     gotoxy(7,7);writeln('Name':20, 'Mark':6, 'No. of Participants':20, 'Mean':6,
'Rank':6);
         for count1 := 1 to schoolnum do
         with School[count1] do
         begin
        gotoxy(7,7+count1);
        writeln(Name:20, Mark:6, Freq:20, Mean:6:1, Rank:6);
         end;
     gotoxy(20,10+count1);
     writeln('Press enter to continue.');
     ch := readkey;
end;
```

```
procedure prize;
begin
             style;
             gotoxy(40,5); write('Prize');
             gotoxy(15,8);write('Individual Awards');
      count2 := 0;

      for count1 := 1 to numofile do
      with student[count1] do
      if rank = 1 then
         begin
           count2 := count2 + 1;
           gotoxy(15, 8 + count2);
           write('Champion       ', name);
         end else if rank = 2 then
         begin
            count2 := count2 + 1;
            gotoxy(15, 8 + count2);
            write('1st runner up  ', name);
         end else if rank = 3 then
         begin
            count2 := count2 + 1;
            gotoxy(15, 8 + count2);
            write('2nd runner up  ', name);
         end;

      gotoxy(15,13);write('School Awards');
      count2 := 0;
      for count1 := 1 to numofile do
      with school[count1] do
      if rank = 1 then
         begin
           count2 := count2 + 1;
           gotoxy(15, 13 + count2);
           write('Champion       ', name);
         end else if rank = 2 then
         begin
            count2 := count2 + 1;
            gotoxy(15, 13 + count2);
            write('1st runner up  ', name);
         end else if rank = 3 then
         begin
            count2 := count2 + 1;
            gotoxy(15, 13 + count2);
            write('2nd runner up  ', name);
         end;

      gotoxy(15,19);
      writeln('Press enter to continue.');
      ch := readkey;
end;
```

‘Prize’ shows which student and which school can get the prizes.

```
procedure exportresult;
var   percent : integer;
      f1, f2, f3 : string;
begin
          str(Year, f1);
          str(Month, f2);
          str(Day, f3)
          assign(Df, ‘E:’+f1+f2+f3+‘Result.txt’);
          rewrite(Df);
   writeln (Df, 'Time: ',DateTimeToStr(Now));
          writeln(Df, 'Performance');
          writeln(Df, 'No. of participants: ', numofile);
   writeln(Df, 'No. of participating schools: ', schoolnum);
   writeln(Df, 'Mean: ',mean :0:1);
   writeln(Df, 'Standard deviation: ', standev :0:1);

          {Output ranking of student}
          writeln(Df, 'Participants'' Performance');
          writeln(Df,'Name':15, 'Mark':6, 'School':15, 'Rank':6);
          for count1 := 1 to numofile do
   with student[count1] do
        writeln(Df, Name :15, Mark :6, School :15, Rank :6);
   writeln(Df);
   writeln(Df, 'School''s Performance');
   writeln(Df, 'Name':15, 'Mark':6, 'No. of Participants':20, 'Mean':6, 'Rank':6);
          for count1 := 1 to schoolnum do
          with School[count1] do
   writeln(Df, Name:15, Mark:6, Freq:20, Mean:6:1, Rank:6);

          writeln(Df);
          writeln(Df, 'Prize');
          writeln(Df, 'Individual Awards');
          for count1 := 1 to numofile do
              with student[count1] do
                     case rank of
              1 : writeln(Df, 'Champion':15, name:15);
                     2 : writeln(Df, '1st runner up':15, name:15);
                     3 : writeln(Df, '2nd runner up':15, name:15);
                     end;

      writeln(Df);
                    writeln(Df, 'School Awards');
                    for count1 := 1 to schoolnum do
                    with school[count1] do
                              case rank of
                              1 : writeln(Df, 'Champion':15, name:15);
                              2 : writeln(Df, '1st runner up':15, name:15);
                              3 : writeln(Df, '2nd runner up':15, name:15);
                              end;

      writeln(Df);
      writeln(Df, 'Question Analysis');
             writeln(Df, 'Question', 'Freq':6, 'Answer': 10, 'Percentage': 13);
             for count1 := 1 to questnum do
                 begin
                                  percent := answerfreq[count1] * 100 div 10;
             writeln(Df, count1:5, answerfreq[count1]:6 , student[0].answer[count1]: 10, percent: 12,'%');
                 end;
          close(Df);
   clrscr;
   gotoxy(31,9);
   writeln('The file is exported!');
   delay(3000);  {UI}
end;
```

```
procedure mainpage;
begin
         style;
     DeCodeDate (Date,Year, Month, Day);
     writeln ('Time: ',DateTimeToStr(Now));
     gotoxy(24,09); writeln('Multiple-Choice Analysis System');
         gotoxy(28,11); writeln('1. Question Analysis');
         gotoxy(28,12); writeln('2. Searching');
         gotoxy(28,13); writeln('3. Performance');
         gotoxy(28,14); writeln('4. Prize');
     gotoxy(28,15); writeln('5. Exporting Result');
         gotoxy(28,16); writeln('Press ''Q'' to exit.');
     gotoxy(28,17);

         ch := readkey;
         case ch of
         '1' : qanalysis;
         '2' : search;
         '3' : performance;
         '4' : prize;
         '5' : exportresult;
         'Q', 'q' : writeln('You now exit the system.');
         else mainpage;
         end;
end;
```

User can press ‘1’ to ‘5’ to choose the function they need and press ‘Q’ or ‘q’ to exit the system but no change for other keys.

‘delay(3000)’ means the screen wait for 3000 milliseconds then go back to main page.

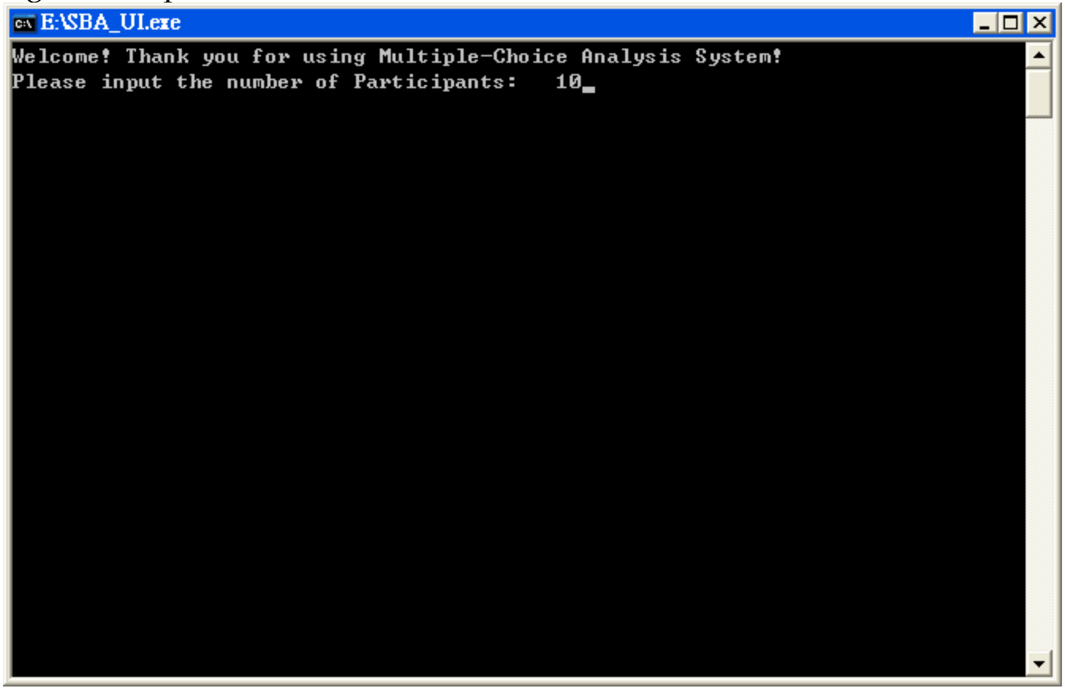
```
begin
      start;
      repeat
                  mainpage;
      until (ch = 'q') or (ch = 'Q');
      READLN;
end.
```

Main program

Program Implementation

Set up

User should input the number of students which is shown on Fig. 2. After that, the system will process all data such as inputing files, marking answers, sorting and finding who can get prizes. All these procedures will not done after the main is shown, it can reduce processing consumption.

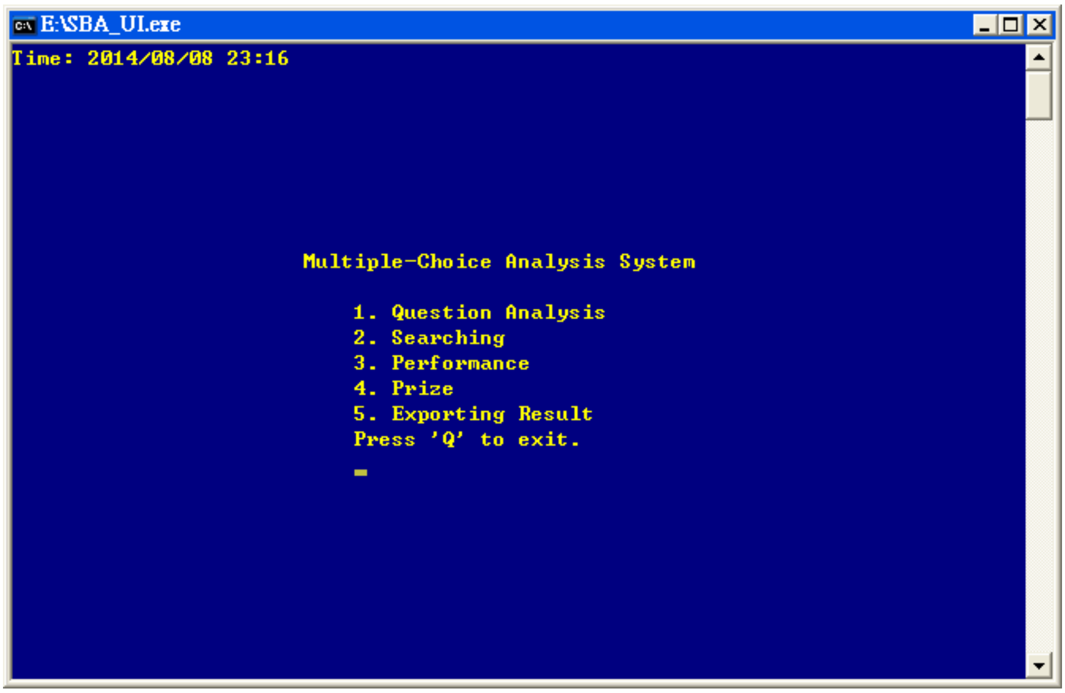


```
Welcome! Thank you for using Multiple-Choice Analysis System!
Please input the number of Participants:   10_
```

Fig. 2 The start of the program

Main Page

After all data are processed, the system will show the main page to user. User just only press key '1', '2', '3', '4' and '5' for different functions. User can press 'Q' to exit the system. Time is shown at the top left corner. To make sure the system can be reusable, it will not save the data inside the software that user don't need to clean up for next time. Dark blue background and bright yellow word show on the screen because it is very comfortable for user to read.

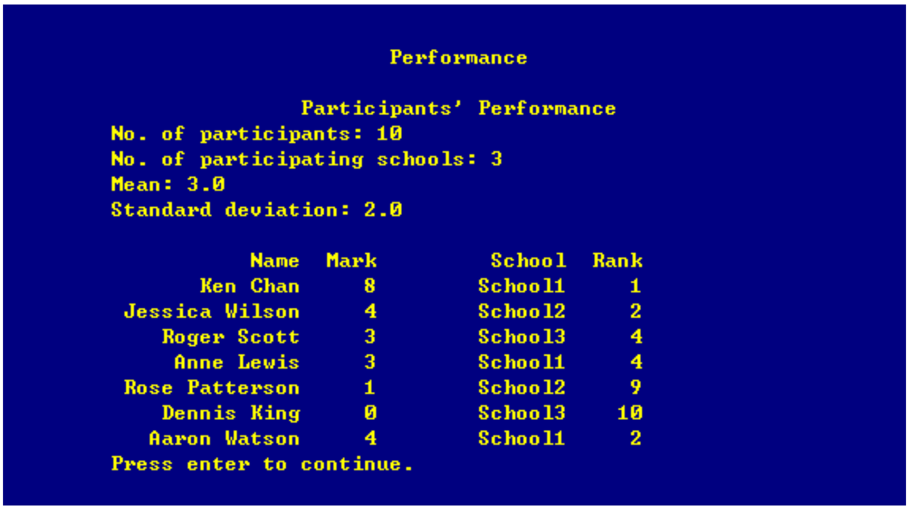


```
Time: 2014/08/08 23:16

                    Multiple-Choice Analysis System

                        1. Question Analysis
                        2. Searching
                        3. Performance
                        4. Prize
                        5. Exporting Result
                        Press 'Q' to exit.
```

Fig. 3 The main page

Performance

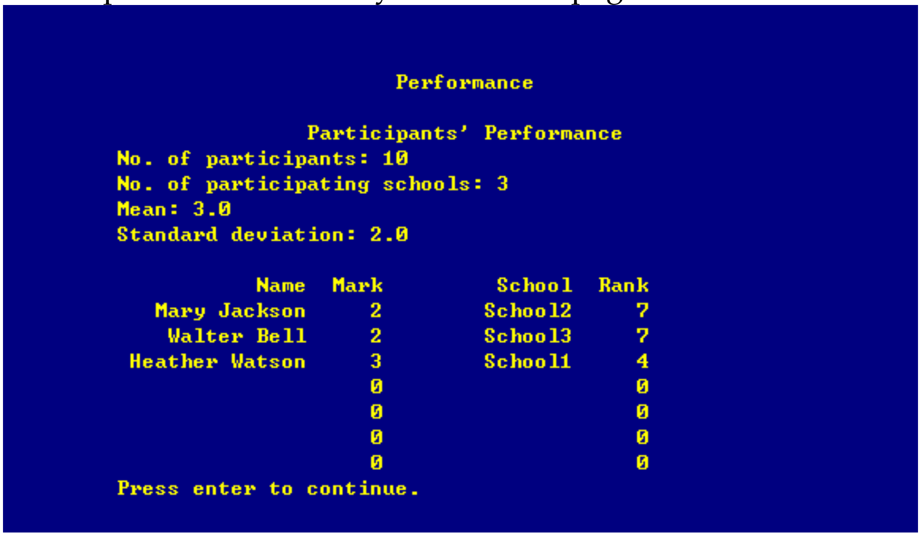
When user press '1', 'Performance' page will be shown. It used to shows marks each students can get, their schools and ranks. Number of participants and participating schools, mean and standard deviation will be shown in this page, that's the first part.



```
                      Performance

               Participants' Performance
No. of participants: 10
No. of participating schools: 3
Mean: 3.0
Standard deviation: 2.0

           Name  Mark         School  Rank
       Ken Chan     8        School1     1
 Jessica Wilson     4        School2     2
    Roger Scott     3        School3     4
     Anne Lewis     3        School1     4
 Rose Patterson     1        School2     9
    Dennis King     0        School3    10
   Aaron Watson     4        School1     2
Press enter to continue.
```

Fig. 4 Student's performance

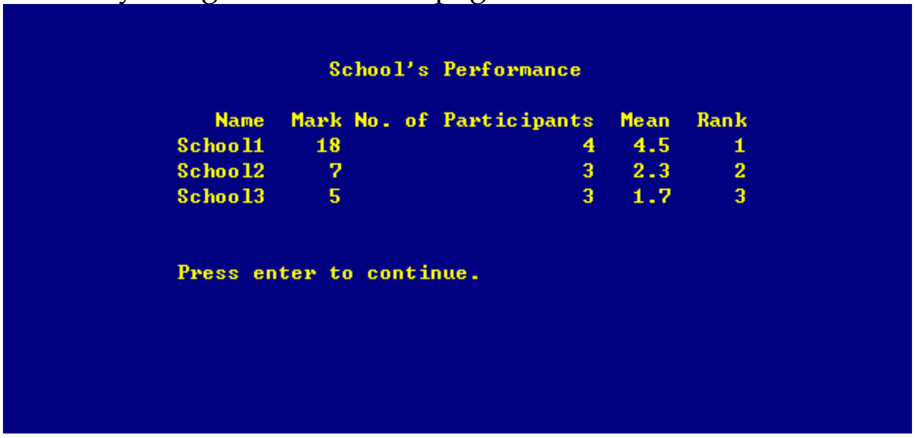
The second part shows all students' mark, rank and school. The system organizes the data very clear and user friendly. Each page only shows 7 records because the screen size is limited, so user can press the 'Enter' key for the next page.



```
                      Performance

               Participants' Performance
No. of participants: 10
No. of participating schools: 3
Mean: 3.0
Standard deviation: 2.0

           Name  Mark         School  Rank
   Mary Jackson     2        School2     7
    Walter Bell     2        School3     7
 Heather Watson     3        School1     4
                    0                    0
                    0                    0
                    0                    0
                    0                    0
Press enter to continue.
```

Fig. 5 Student's performance on the second page

The third part is the performance of each school. It shows number of participants, marks rank. Press 'Enter' key can go back to main page.

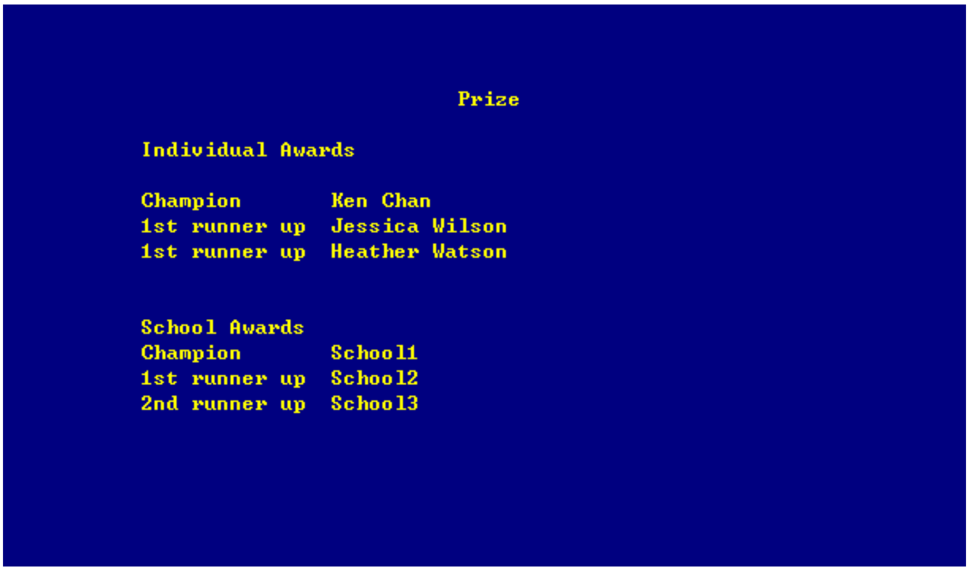


```
            School's Performance

   Name  Mark No. of Participants  Mean  Rank
School1    18                   4   4.5     1
School2     7                   3   2.3     2
School3     5                   3   1.7     3


Press enter to continue.
```

Fig. 6 School's performance

Prize

This is 'Prize'. A very clean page which show which students and schools can get Champion, 1st runner and 2nd runner. The analysis has already done before the main is shown. If there are more than one participants get the same mark, they will get the same rank, but the next rank will not appear. From the screen, there are two first runner up but no second runner up. Before the start of this function, the system already have done all calculation, this procedure only show the information. That can make sure the system can run faster.

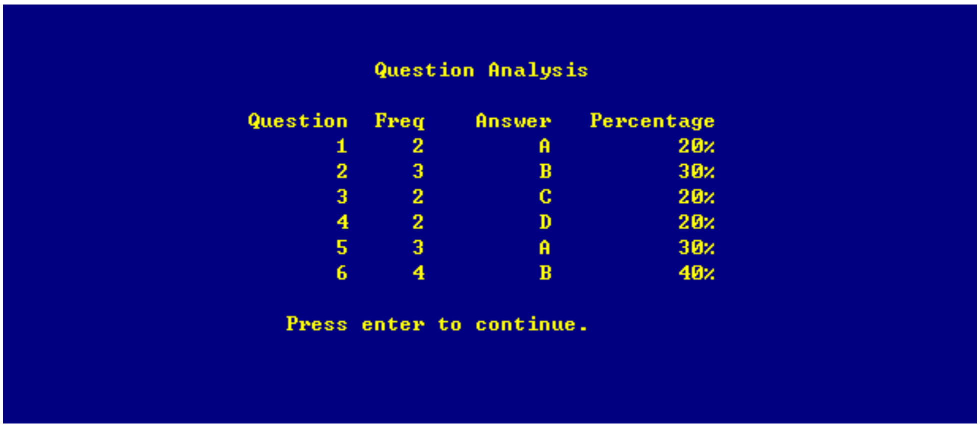


```
                         Prize

Individual Awards

Champion        Ken Chan
1st runner up   Jessica Wilson
1st runner up   Heather Watson


School Awards
Champion        School1
1st runner up   School2
2nd runner up   School3
```

Fig. 7 Individual Awards and School Awards

Question Analysis

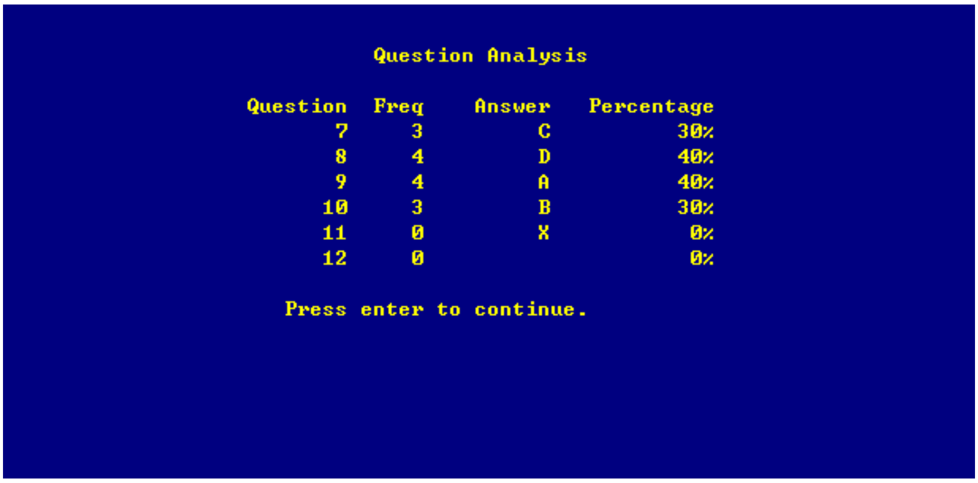
Question Analysis shows student overall performance in every question. The percentage of each question is shown next to the answer, user can see which question is the hardest or easiest.



```
                Question Analysis

   Question  Freq     Answer   Percentage
          1     2          A          20%
          2     3          B          30%
          3     2          C          20%
          4     2          D          20%
          5     3          A          30%
          6     4          B          40%

      Press enter to continue.
```

Fig. 8 The first page of Question Analysis

Again, because of the screen size, it only shows 6 records in one page and press enter to go through the rest of record. This is a good reference for teachers to know participants' strength.

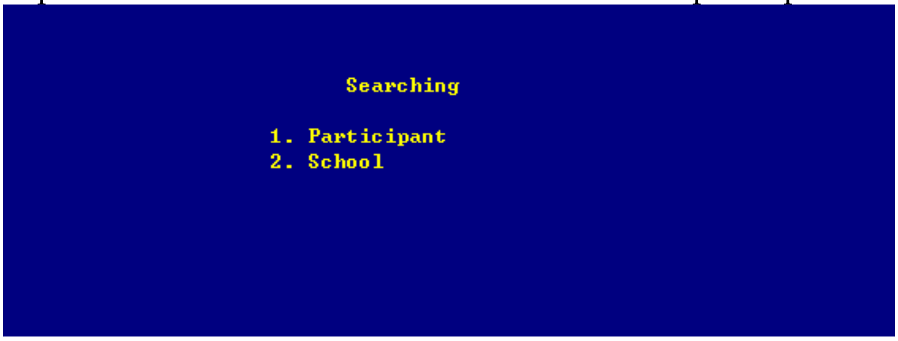


```
                Question Analysis

   Question  Freq     Answer   Percentage
          7     3          C          30%
          8     4          D          40%
          9     4          A          40%
         10     3          B          30%
         11     0          X           0%
         12     0                      0%

      Press enter to continue.
```

Fig. 9 The second page of Question Analysis

Searching

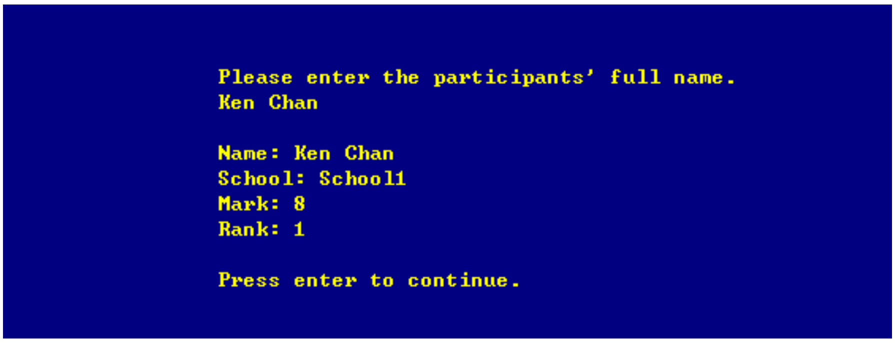
User can press ‘1’ or ‘2’ to find the information for each participant or school.



```
      Searching

 1. Participant
 2. School
```

Fig. 10 The choices for searching

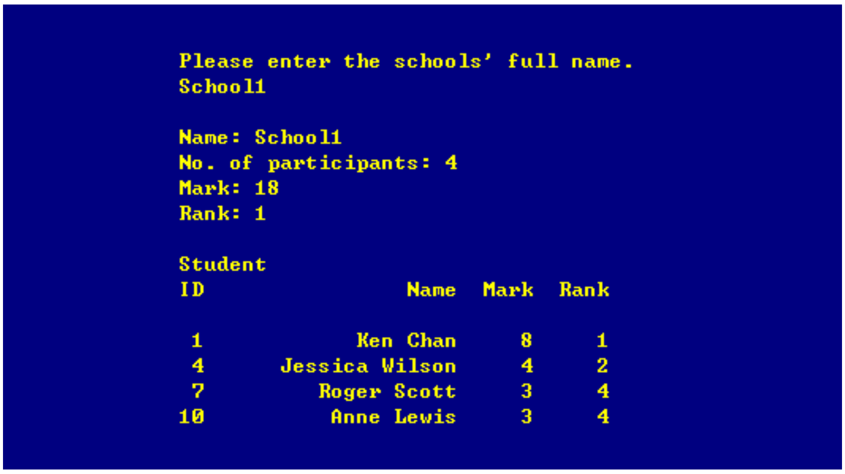
The program will not do anything if user press any key except ‘1’ and ‘2’ to make sure the program will not get any error.



```
Please enter the participants' full name.
Ken Chan

Name: Ken Chan
School: School1
Mark: 8
Rank: 1

Press enter to continue.
```

Fig. 11 Showing the student’s information

When user press ‘2’, user can input the school’s name and find the information about them, it is show the number if participants, mark and rank for specific school. At the lower half of page, participants who are studying in the school will be appear and each student’s mark and rank will be shown.

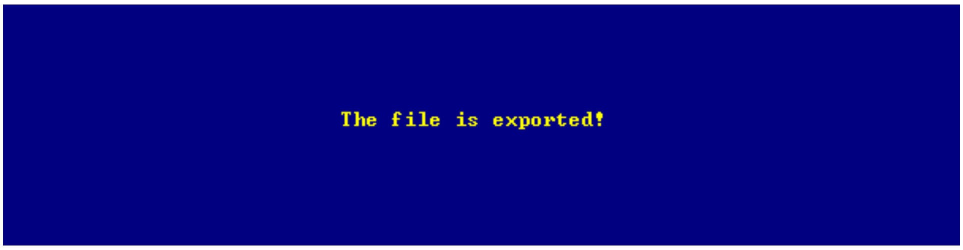


```
Please enter the schools' full name.
School1

Name: School1
No. of participants: 4
Mark: 18
Rank: 1

Student
ID                Name  Mark  Rank

 1            Ken Chan     8     1
 4      Jessica Wilson     4     2
 7         Roger Scott     3     4
10          Anne Lewis     3     4
```

Fig. 12 Showing the school’s information

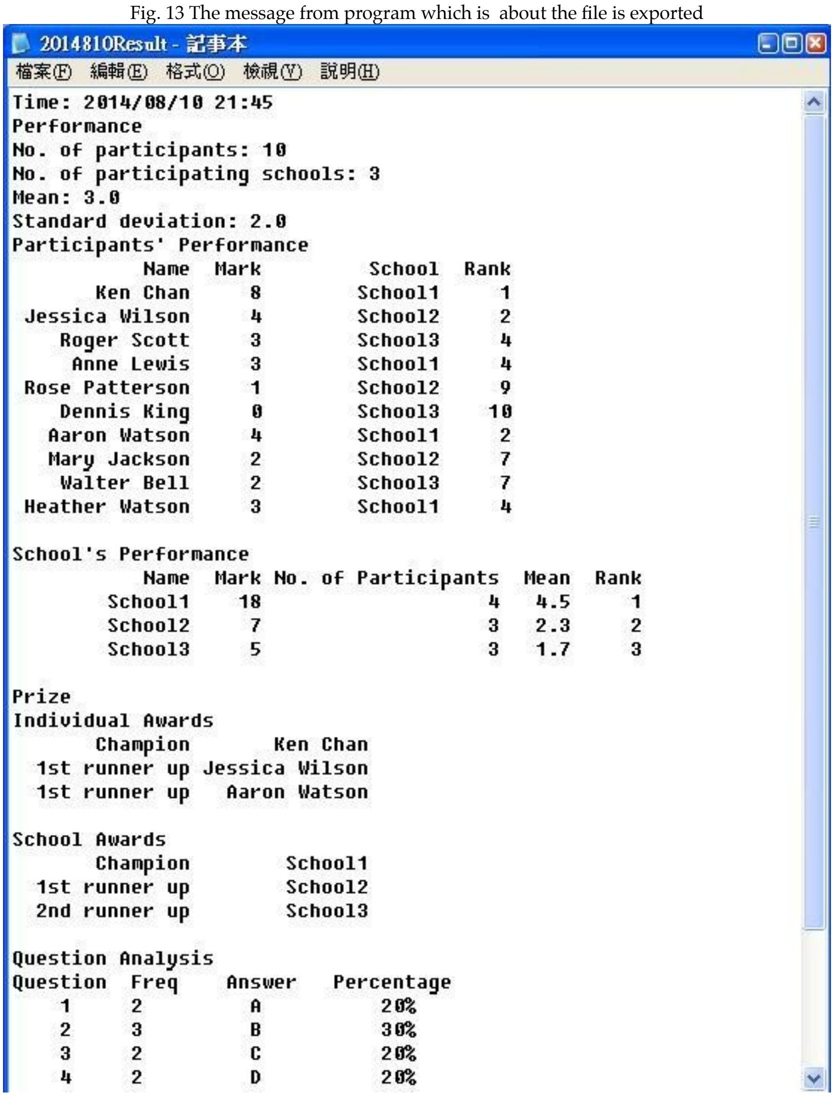
Exporting Result

A file ‘xxxxResult.txt’ will be exported when user press ‘5’ in the main page. The program will show the message ‘The file is exported!’ about 5 seconds and go back to main page. In‘xxxx’ means the date of outputting the result. It helps the computer won’t recover the old file and easy to organize files. The file contain all data such as overall performance for each student. The purpose for exporting file is to help user to backup the data and print the result if they need. Also text file is system independent, user can open the file by using different operating system.



```
The file is exported!
```

Fig. 13 The message from program which is about the file is exported



```
Time: 2014/08/10 21:45
Performance
No. of participants: 10
No. of participating schools: 3
Mean: 3.0
Standard deviation: 2.0
Participants' Performance
           Name  Mark         School  Rank
       Ken Chan     8        School1     1
 Jessica Wilson     4        School2     2
    Roger Scott     3        School3     4
     Anne Lewis     3        School1     4
 Rose Patterson     1        School2     9
    Dennis King     0        School3    10
   Aaron Watson     4        School1     2
   Mary Jackson     2        School2     7
    Walter Bell     2        School3     7
 Heather Watson     3        School1     4

School's Performance
           Name  Mark No. of Participants  Mean  Rank
        School1    18                   4   4.5     1
        School2     7                   3   2.3     2
        School3     5                   3   1.7     3

Prize
Individual Awards
       Champion       Ken Chan
  1st runner up Jessica Wilson
  1st runner up   Aaron Watson

School Awards
       Champion        School1
  1st runner up        School2
  2nd runner up        School3

Question Analysis
Question  Freq    Answer   Percentage
    1     2         A          20%
    2     3         B          30%
    3     2         C          20%
    4     2         D          20%
```

Fig. 14 The exported file

Testing

Testing plan

The testing will test the following functions of the program. The table shows the case will be appeared and the expected result.

Set up	Expected Result
Ordinary case	Pass
Invalid data	Shows Warning message
Searching	
Ordinary case	Pass
Invalid data	Shows Warning Message
Performance, Prize and Exporting Result	
No repeat mark	Pass
More than one get the same item	Pass
Question Analysis	
Add answer	Pass
Delete answer	Pass
Change answer	Pass

Setting

Before the system input the data, user need to input number of students. When user press '0' or non-numeral characters in number of students, there will appear a message "Input error" which tells user input the data again.

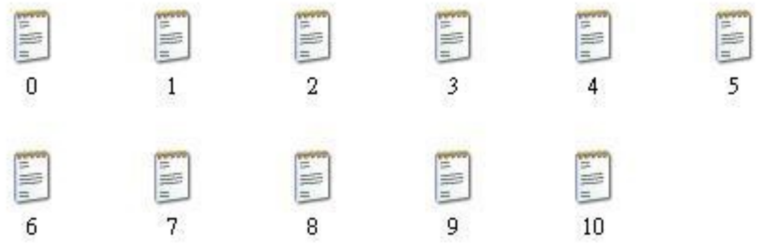
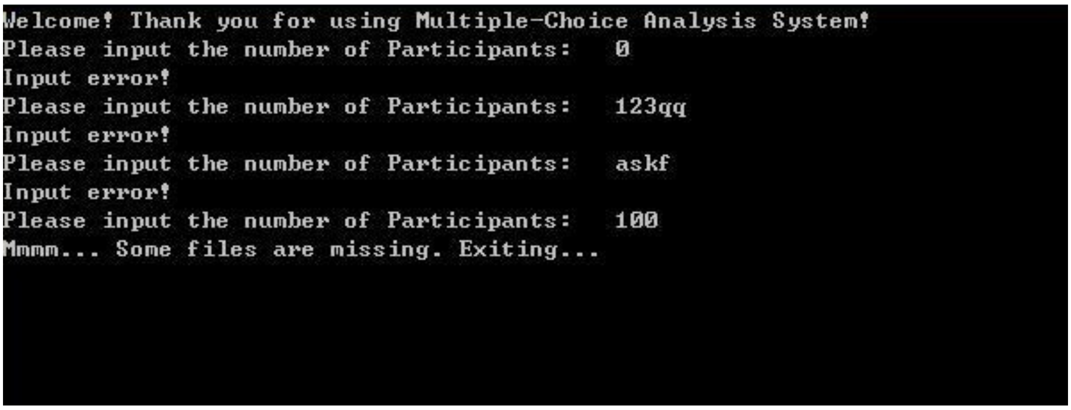


Fig. 15

Let's say there are 10 students join this competition. As I said, each file represent each student. When user type English characters and symbols, the system will not accept will and it will ask user to input again. When user input more than 10 students or some files are missed, the system will give a message 'Mmmm… Some files are missing. Exiting…'. User press any key, the system will be shut down and user need to checks is there any files are missed.(0.txt means the model answers.)

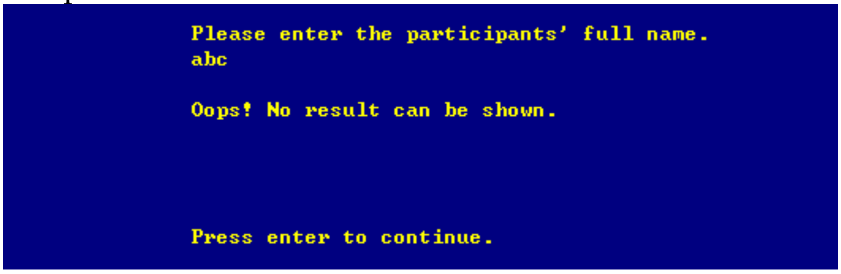


```
Welcome! Thank you for using Multiple-Choice Analysis System!
Please input the number of Participants:   0
Input error!
Please input the number of Participants:   123qq
Input error!
Please input the number of Participants:   askf
Input error!
Please input the number of Participants:   100
Mmmm... Some files are missing. Exiting...
```

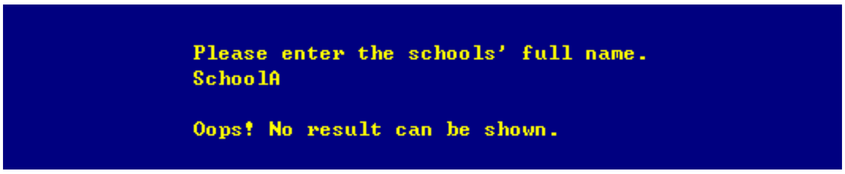
Fig. 16 Input some invalid data and the program shows error message

Searching

User should should need to input student's or school's whole name. Then user need to press enter and go back to main page. In fact, there are no school is called 'SchoolA' and no participant is called 'abc'. If the program can't find any name is the same as the target, there will be a message 'Oops! No result can be shown.'. If the program can find the target, it will show the information that relate to user. The example have been shown in the 'Implement' chapter.



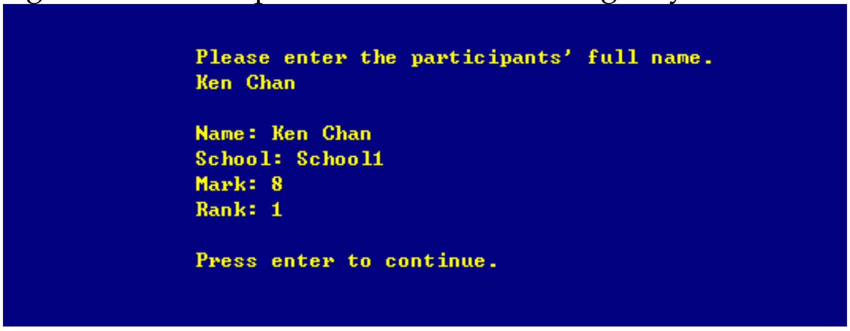
```
Please enter the participants' full name.
abc

Oops! No result can be shown.




Press enter to continue.
```



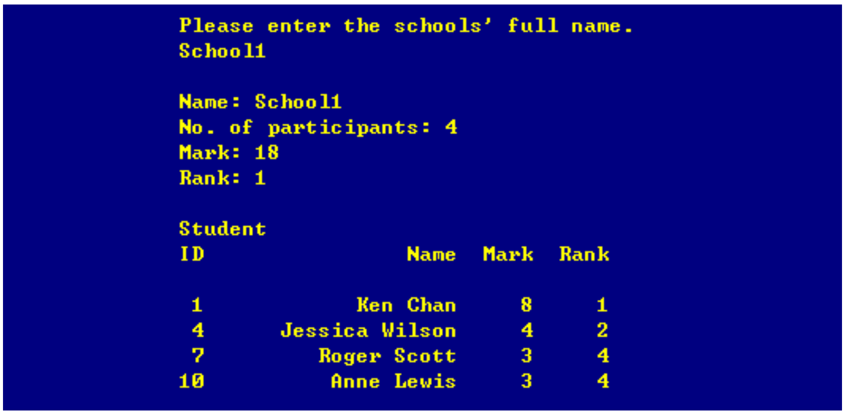
```
Please enter the schools' full name.
SchoolA

Oops! No result can be shown.
```

Fig. 17 & Fig. 18 Error message in Searching

When the program can find the target, the information will show to user, the first one is to search specific participants and the second one is to search schools' information and some participants' information will be shown is they study in the school. There is no any error can mark the program crash and process the data in wrong way.



```
Please enter the participants' full name.
Ken Chan

Name: Ken Chan
School: School1
Mark: 8
Rank: 1

Press enter to continue.
```



```
Please enter the schools' full name.
School1

Name: School1
No. of participants: 4
Mark: 18
Rank: 1

Student
ID                  Name  Mark  Rank

 1              Ken Chan     8     1
 4        Jessica Wilson     4     2
 7           Roger Scott     3     4
10            Anne Lewis     3     4
```

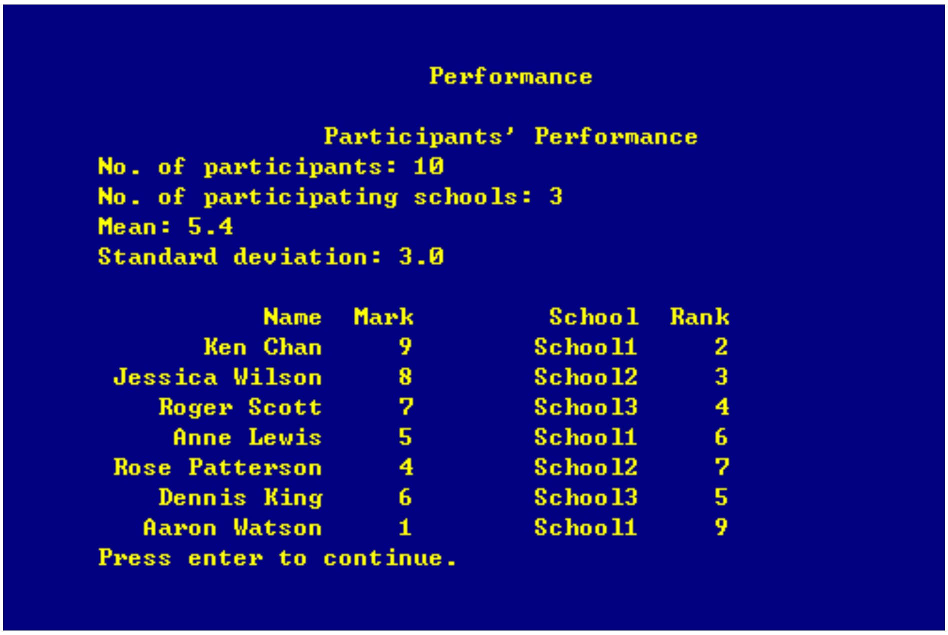
Fig. 19 & Fig. 20 Valid data

Performances, Prize and Exporting Result

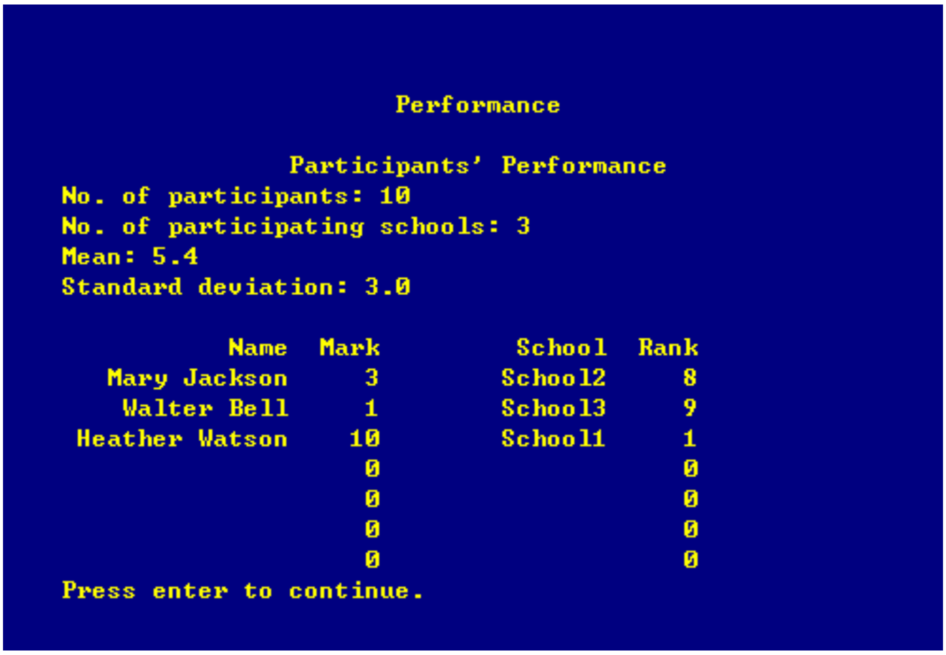
Case 1: None of participant and school get the same prize

This 3 functions test together because they share the same database, which is highly dependent. If some data get error, all of them will get wrong information. It is better than test separately.

There are 10 students, datum in number of participants, number of participating schools, mean and standard deviation are correct. Only 2 students get the same mark and get the same rank.



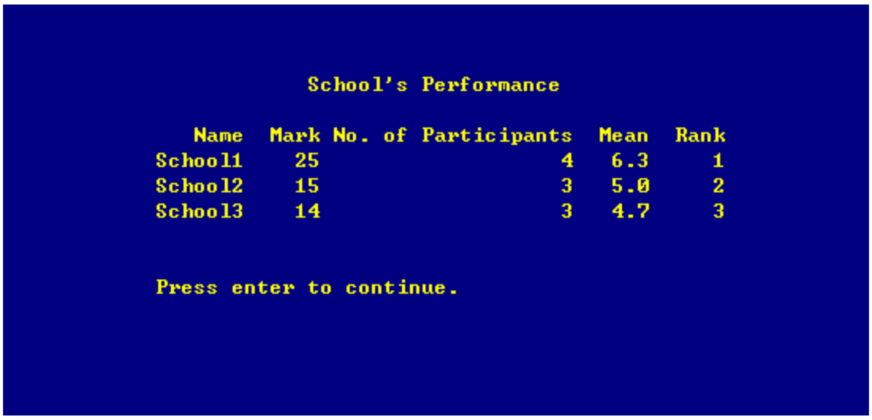
```
                          Performance

                   Participants' Performance
    No. of participants: 10
    No. of participating schools: 3
    Mean: 5.4
    Standard deviation: 3.0

               Name  Mark          School  Rank
           Ken Chan     9         School1     2
     Jessica Wilson     8         School2     3
        Roger Scott     7         School3     4
        Anne Lewis      5         School1     6
     Rose Patterson     4         School2     7
        Dennis King     6         School3     5
       Aaron Watson     1         School1     9
    Press enter to continue.
```



```
                          Performance

                   Participants' Performance
    No. of participants: 10
    No. of participating schools: 3
    Mean: 5.4
    Standard deviation: 3.0

               Name  Mark          School  Rank
       Mary Jackson     3         School2     8
        Walter Bell     1         School3     9
     Heather Watson    10         School1     1
                        0                     0
                        0                     0
                        0                     0
                        0                     0
    Press enter to continue.
```

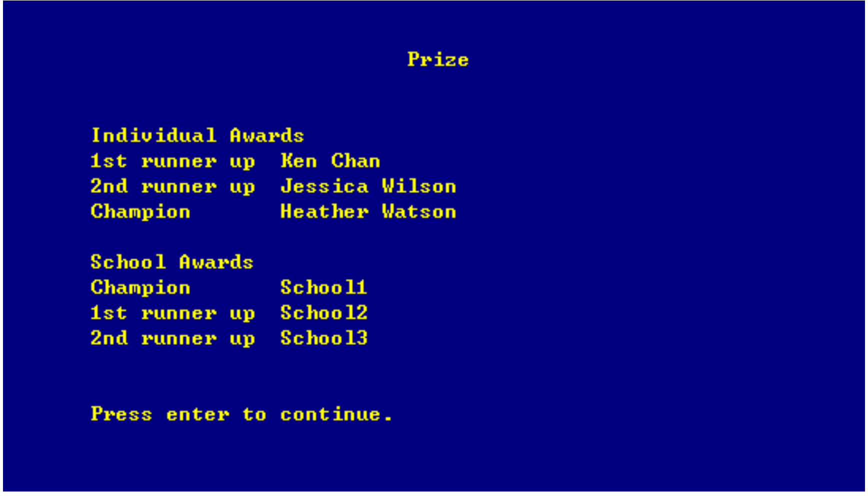
Fig. 21 & Fig. 22 Student's performance

All data in 'School's performance' are correct include mark, number of participants, mean and rank. The system the right prize to specific participants and schools.



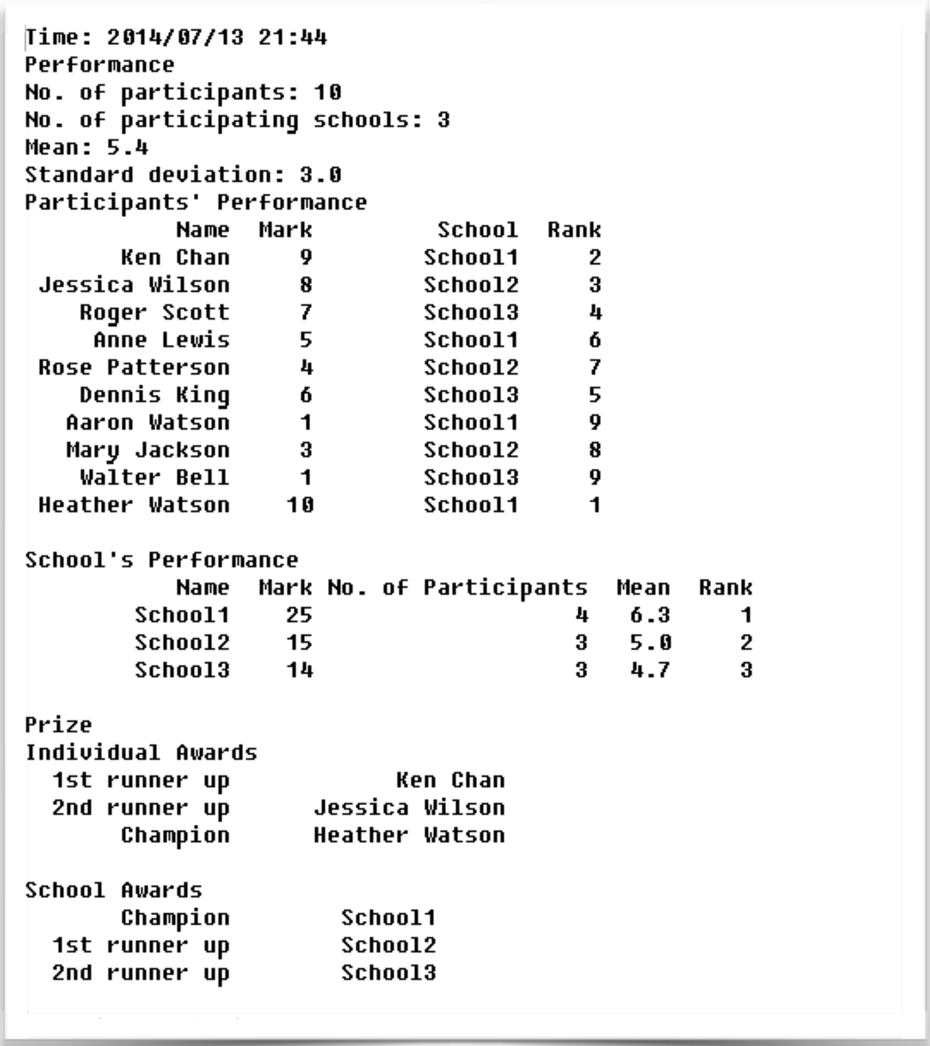
```
                    School's Performance

           Name  Mark No. of Participants  Mean  Rank
        School1    25                   4   6.3     1
        School2    15                   3   5.0     2
        School3    14                   3   4.7     3


        Press enter to continue.
```

Fig. 23 School's Performance



```
                           Prize


      Individual Awards
      1st runner up  Ken Chan
      2nd runner up  Jessica Wilson
      Champion       Heather Watson

      School Awards
      Champion       School1
      1st runner up  School2
      2nd runner up  School3


      Press enter to continue.
```

Fig. 24 Prize



```
Time: 2014/07/13 21:44
Performance
No. of participants: 10
No. of participating schools: 3
Mean: 5.4
Standard deviation: 3.0
Participants' Performance
           Name  Mark         School  Rank
       Ken Chan     9        School1     2
 Jessica Wilson     8        School2     3
    Roger Scott     7        School3     4
     Anne Lewis     5        School1     6
 Rose Patterson     4        School2     7
    Dennis King     6        School3     5
   Aaron Watson     1        School1     9
   Mary Jackson     3        School2     8
    Walter Bell     1        School3     9
 Heather Watson    10        School1     1

School's Performance
           Name  Mark No. of Participants  Mean  Rank
        School1    25                   4  6.3     1
        School2    15                   3  5.0     2
        School3    14                   3  4.7     3

Prize
Individual Awards
  1st runner up            Ken Chan
  2nd runner up      Jessica Wilson
       Champion      Heather Watson

School Awards
       Champion        School1
  1st runner up        School2
  2nd runner up        School3
```

Fig. 25 The exported result shows all information is correct

As the result, there are no any problem in exporting result.

Case 2: More than 1 participants and schools get the same prizes

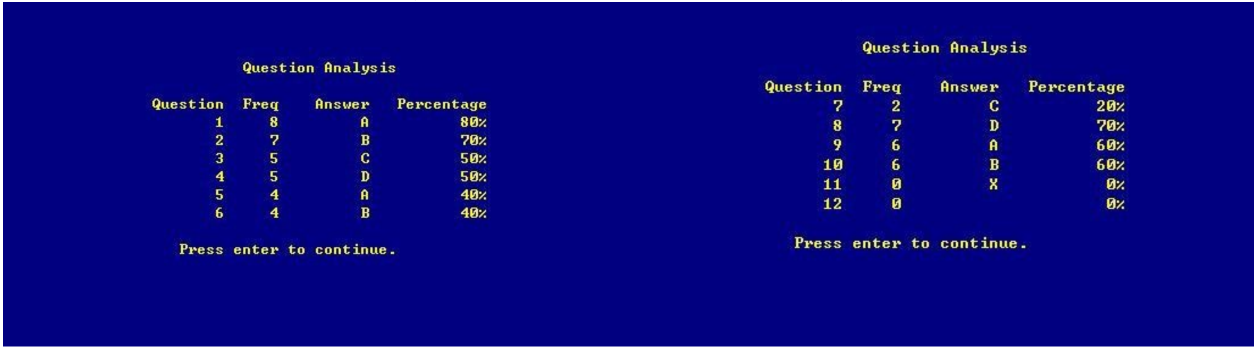
In case 2, the result has ready shown in ‘Implement’ chapter, when there are more than 1 participants get the same mark, the rank will be the same and the next rank will be skipped. So than there are only two first runner up but no second runner up. After exporting report, it shows that same information will performance and prize.

Question Analysis

There are 10 questions, all students' answer are shown in the table. On the screen, it shows the frequency of each question, the modal answer and percentage of students can get right answers.

Case 1: 10 participants

Students' Answer	CBADACBDAB	ABCCABCDAB
	BADCDBACDC	ABCDABBDAC
	ACBCDABDCB	ABCDCABDAB
	ACBADCABDC	ABCACDBAAB
	ABCDABCDAB	ABDDBCAD



```
            Question Analysis

Question   Freq     Answer    Percentage
       1      8          A           80%
       2      7          B           70%
       3      5          C           50%
       4      5          D           50%
       5      4          A           40%
       6      4          B           40%

       Press enter to continue.
```

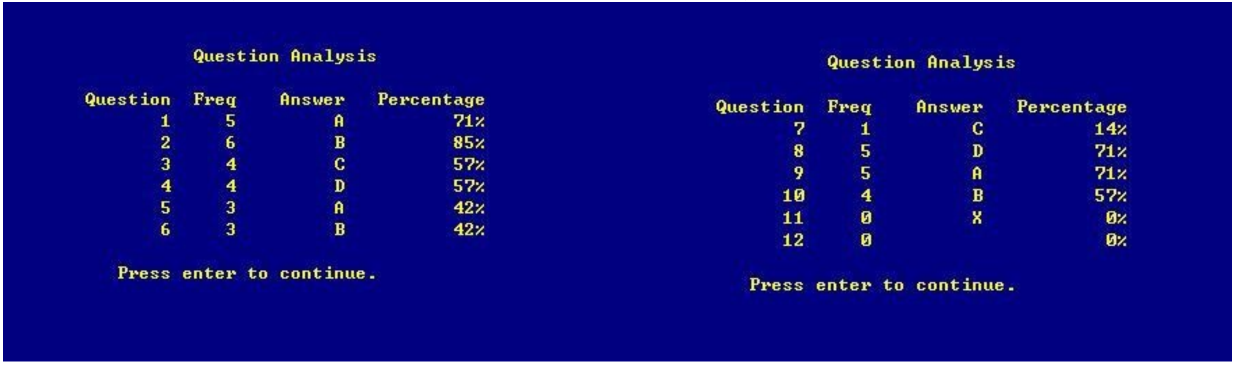
```
            Question Analysis

Question   Freq     Answer    Percentage
       7      2          C           20%
       8      7          D           70%
       9      6          A           60%
      10      6          B           60%
      11      0          X            0%
      12      0                       0%

       Press enter to continue.
```

Fig. 26 & Fig. 27 Question Analysis in case 1

Each page can only show 6 questions because the limitation of space, user can press enter to continue. There is no problem in 'Question Analysis'.

Case 2: 7 participants

Modal Answer	ABCDABCDAB			
Students' Answer	ABCCABCDAB	CBADACBDAB	ABCDCABDAB	ABDDBCAD
	ABCDABBDAC	BADCDBACDC	ABCACDBAAB	



```
            Question Analysis

Question   Freq     Answer    Percentage
       1      5          A           71%
       2      6          B           85%
       3      4          C           57%
       4      4          D           57%
       5      3          A           42%
       6      3          B           42%

       Press enter to continue.
```

```
            Question Analysis

Question   Freq     Answer    Percentage
       7      1          C           14%
       8      5          D           71%
       9      5          A           71%
      10      4          B           57%
      11      0          X            0%
      12      0                       0%

       Press enter to continue.
```

Fig. 28 & Fig. 29 Question Analysis in case 2

In case 2, all information are correct and the typesetting do not affect of the number of data. In conclude, the testing shows there is no any problem in this function.

Summary

Set up	
Ordinary case	Passed
Invalid data	Showed warning message/exit the program
Searching	
Ordinary case	Passed
Invalid data	Showed warning message/exit the program
Performance, Prize and Exporting Result	
No repeat mark	Passed
More than one get the same item	Passed
Question Analysis	
Add answer	Passed
Delete answer	Passed
Change answer	Passed

Evaluation

Targets	Completion(%)
Clear, user-friendly interface	80
Searching	80
Data Processing	100
Data Organization	100
Prize	100
Data Output	100

As the table shown, most of the targets can achieve 100% completion such as data processing and data organization. User input different data, the program will show correct information for user. The special features in this program is it divided into two parts, one part is processing data and another is showing information. It is very great because computer don't need to do many calculations, not just it has user-friendly interface, the structure of the code is also very clear.

Although some procedures are only run once only, but this method can reduce energy consumption as computer don't need to calculate something before showing data. The program cannot provide completely clear, user-friendly interface because it is written by Pascal, which do not support mouse to click buttons and not very attractive. For searching, user just need to provide surname and type just type small latter or capital letter to increase efficiency, program cannot do that because it will spend so much time to write code. Sorting will be very slow if large amount of data are need to process, quick sort can be used but it is very difficult to understand the idea and the process.

Although there are some problem appear in this program, but it doesn't make highly bad effect for user. The program's advantages are much more than the disadvantages.

Conclusion & Discussion

Multiple choice analysis system is a very great software for testing, teachers don't need to mark and do any calculating so the error can be reduced. There are many useful functions in this program such as marking, searching, showing the performance and outputting the result. Mainly there is no problems and all data are processed properly, also all input will not crash down the system as many protections are considered. User can safely use this program.

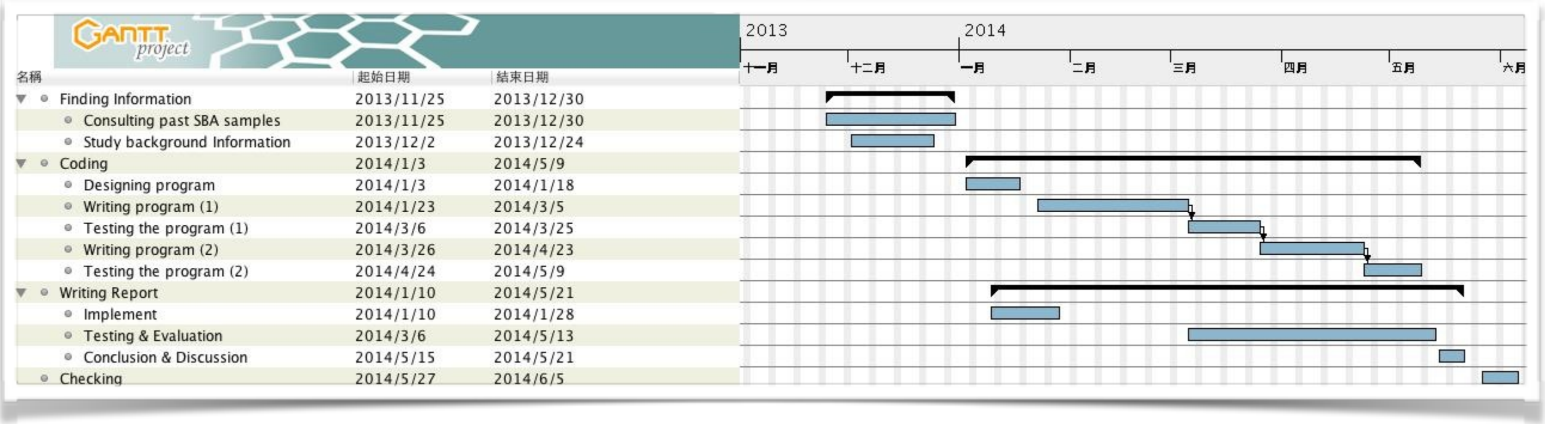
After finishing the program, it is hard to believe the program can be run perfectly. Before program coding, a lot of time are spent to do research and studied past SBA to deter main the direction of programming. Gantt graph are designed for time management. All of these hard work are wanted to make a good start.

Many skills are learned from ICT lessons helped a lot. For finding information, searching by Google to find solutions to solve problems. Thanks for daily lessons, flow chart can be used to express own idea. Google Drive and Microsoft Word are used for editing report with some advance functions such as cover function, header and page number can, and table. The most interesting thing in ICT SBA is something cannot learn in daily lessons, someone published on interesting method to solve some specific problems. It is different from other subjects, there is no model answer to follow, programmer should do all things by their own.

Although this program is very great, there should be some improvements to enhance the program's quality.

1. ***More user-friendly***
   Pascal cannot used by mouse and insert pictures but C++, Visual Basic and Java can. Also user can click buttons without typing commands by using Graphical User Interface.
2. ***Pie chart and bar chart***
   It can be provided in the system as it involved many statistic operating. But Pascal do not have strong graphic strength, so using other computing languages as mentioned before can improve this situation, data can be shown more straight forward.
3. ***Support PDF, MS Word and MS Excel***
   These file types are very popular because they have many useful features that text file cannot provide such as insert pictures, change fonts, add tables. User can share the report much more convenience.
4. ***Support Windows, Mac and Linux***
   It is convenient for user to process data by using different operating system. But the file type of the system is .exe, which Mac OS and Linux do not support it, less user can use the program.

Project management



名稱	起始日期	結束日期
Finding Information	2013/11/25	2013/12/30
Consulting past SBA samples	2013/11/25	2013/12/30
Study background Information	2013/12/2	2013/12/24
Coding	2014/1/3	2014/5/9
Designing program	2014/1/3	2014/1/18
Writing program (1)	2014/1/23	2014/3/5
Testing the program (1)	2014/3/6	2014/3/25
Writing program (2)	2014/3/26	2014/4/23
Testing the program (2)	2014/4/24	2014/5/9
Writing Report	2014/1/10	2014/5/21
Implement	2014/1/10	2014/1/28
Testing & Evaluation	2014/3/6	2014/5/13
Conclusion & Discussion	2014/5/15	2014/5/21
Checking	2014/5/27	2014/6/5

Fig. 30 The Gantt Chart

The project is operated from end November of 2013 to early June of 2014, which is divided into 4 parts: finding information, coding, writing report and checking.

During the project, I code in most of the time, it spent more than 4 months because it is the most important in the project. Most of the tasks in coding are dependent tasks as tasks should be finished step by step. Also there are some concurrent tasks during coding such as writing report, so that time schedule is very important to complete different tasks. Overall, the project works smoothly although there are some problems cause some tasks cannot finish on time.

Gantt chart is very important for me because it allows me to make a realistic assessment on the progress and the completion time of this project, shows the dependencies of various activities in this project and easily shows concurrent activities. In the future, I will need to lead a team to finish a project, I cannot concentrate on the resources to be allocated to each activity without Gantt Chart.

Reference

User interface - Wikipedia
http://en.wikipedia.org/wiki/User_interface

Multiple Choice - Wikipedia
http://en.wikipedia.org/wiki/Multiple_choice

Pascal Programming
http://pascal-programming.info/index.php

Gantt chart
http://www.gantt.com/index.htm

Mac OS X
http://www.apple.com/osx/

Cover Picture
http://3.bp.blogspot.com/-LhDFJJsSilU/Uq2FRv3eu2I/AAAAAAAADy0/QvLrSkTL2qY/s1600/multiple-choice-test.jpg

End