

# **Hong Kong Diploma of Secondary**

**Education Examination 201X**

**Information and Communication Technology**

**School-based Assessment**

# Table of contents

<b>Chapter 1 - Introduction .....</b>	3
1.1    Background .....	3
1.2    Objectives .....	4
1.3    Project Plan .....	6
<b>Chapter 2 – Analysis .....</b>	8
2.1    Competition Regulations .....	8
2.2    Data collection .....	9
2.3    Input, Output and Process .....	10
2.4    Choice of IT Tools .....	13
<b>Chapter 3- Design of Solution .....</b>	21
3.1    Brief Description .....	21
3.2    Refinement of Problem.....	21
3.2    Input data method and file formats.....	24
3.3    Output Format .....	28
<b>Chapter 4 Implementation .....</b>	42
4.1    Brief Description .....	42
4.2    Procedures in the Program .....	42
4.3    Data Structures .....	77
4.4    Program Coding.....	83
4.4    Program Execution.....	84
<b>Chapter 5 Testing &amp; Evaluation .....</b>	98
5.1    Brief Description .....	98
5.2    Testing and Evaluation Plan .....	98
5.3    Internal testing.....	99
5.4    Compatibility testing.....	110
5.5    Debugging after testing .....	112
5.4    Self-Evaluation .....	116
5.5    External Testing and Evaluation .....	117
<b>Chapter 6 Conclusion &amp; Discussion .....</b>	118
6.1    Pros and cons of the program.....	118
6.2    Future Improvement.....	119
6.3    Self-Reflection .....	120
<b>Chapter 7 Reference and Acknowledgement .....</b>	121
Reference .....	121
Acknowledgement .....	122

## Multiple-Choice Analysis Report

<b>Appendices .....</b>	123
Appendix I - Program Code .....	123
Appendix 2 - Working schedule .....	152



# **Chapter 1 - Introduction**

## **1.1 Background**

An inter-school ICT competition, in the form of multiple-choice questions, has been organized. The answer sheets for participants will be read by using an Optical Mark Recognition (OMR) system, and the options selected were converted into a **text file**.

My job is to develop a computer program that can read all **raw text-based data** and produce a **detailed analysis report** on the competition

Reading data from the competition

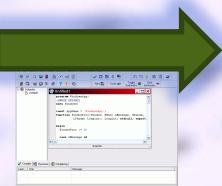


Interpreting and analysing the data

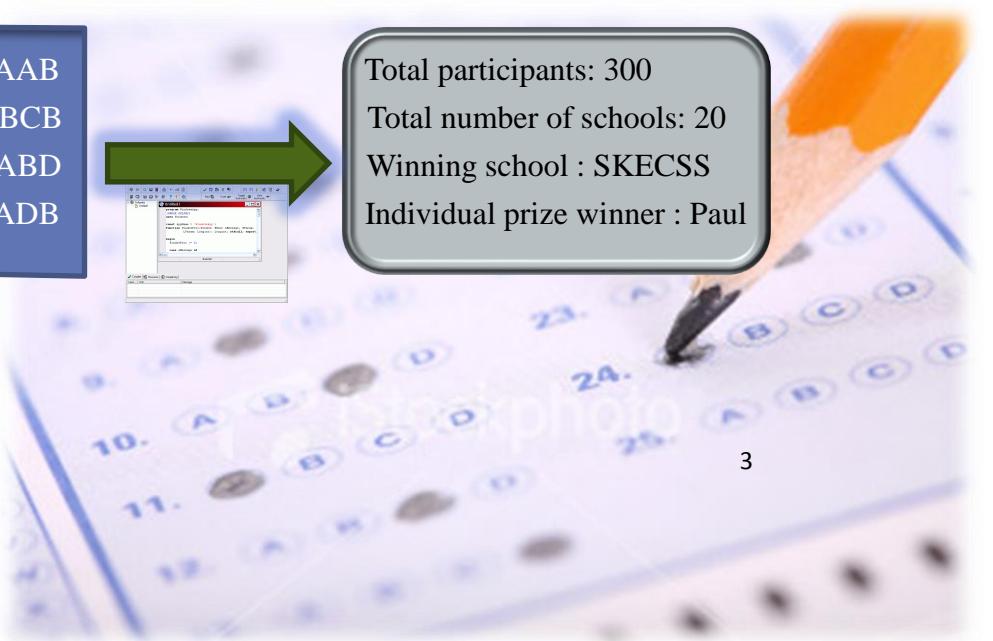


Generating a detailed analysis report

Alvin ABDDCCDDDBAAAAB  
Bob ABCDDBCBDDAABC  
Jacky BAABDDBCDDDBABD  
Jenny ABBDDBCDDBABADB



Total participants: 300  
Total number of schools: 20  
Winning school : SKECSS  
Individual prize winner : Paul



## 1.2 Objectives

The analysis report generated should include:

- ⌚ Total number of participants
- ⌚ Total number of participating schools
- ⌚ Total number of participant(s) from each participating school
- ⌚ Winners of individual awards and school awards
- ⌚ Question analysis, such as the percentage correct for each question
- ⌚ Statistics of the results

In simple words, a program that can

- ⌚ Read all raw text-based data from the answer sheets
- ⌚ Analyze the data and work out different measures of statistics from the answer sheets.
- ⌚ Generate an analysis report

should be generated.

For this, sub-problems that the program should be able to solve are:

- ⌚ To read the answer and the information of participants
- ⌚ To read the answer key for the questions
- ⌚ To compare the answers of participants to the answer key
- ⌚ To calculate the score of each participant
- ⌚ To calculate the results of each school
- ⌚ To store these data
- ⌚ To sort the records for the ranking of participants
- ⌚ To find out the award winners
- ⌚ To analyze questions and calculate for the statistics
- ⌚ To generate a report on visual display unit
- ⌚ To save the report in text form for users

### Identification of users' requirements:

The users of this program will be the **organizers** and **analysts** of the competition.

They would like to have the **list of award winners** and **number of participants**, they need the program to count and compare to find out those information.

## Multiple-Choice Analysis Report

Also, they would like to know the **general performance of students**, such as the **percentage correct** for each question to adjust the difficulty of the competition in the next year, and so **statistical information** such as percentage correct as mentioned, **standard deviation**, **mean score** and **median score** should also be included in the analysis report.

The program should be designed as **user-friendly** as possible to suit all kinds of users. **Prompt words** for users should be very **clear** to indicate the data that have to be inputted by the users. Also, the users should be able to generate the report in a **few steps** to enhance efficiency.

Also, the program should be designed to be used on **personal computers**, which most users may have at home or at their working places

## Functions / features for users

- ⌚ Provide the list of the **award winners**( e.g. schools getting champion, first runner-up, second runner-up, individual award winners in each school)
- ⌚ Provide the list of **numbers of participants** in each school / in total.
- ⌚ Provide **statistical information about students' performance**, such as:  
Standard deviation of correct answers in each question, median score/ mean score of each question and in the whole competition. Percentage correct of each question, the number of each answers chosen in each question)

## Key concepts involved

On Programming:

- ⌚ Using **arithmetic operators** to calculate for the competition results
- ⌚ Using different **data structures** like arrays and records to store the data
- ⌚ Using different **selection algorithms** to search for the designated record
- ⌚ Using different **sorting algorithms** to sort the data, like ranking, finding maximum and minimum value of marks
- ⌚ Using **iterations** when repeated tasks or large amount of data is involved
- ⌚ Using **text manipulation** skills for making analysis report in text form
- ⌚ Using **program library** to call for advanced operations

On results analysis:

- ⌚ Calculations for a **measure of central tendency**, like mean, median, mode

- ⌚ Calculations for a **measure of dispersion**: like range, inter-quartile range, standard deviation

## 1.3 Project Plan

The followings are the sub-divided steps to accomplish the missions of the program:

### Analysis:

- ⌚ Define the competition **regulations**, such as how many choices for each question, the criteria for giving the prizes to individual and school awards winners
- ⌚ Identify the input( **what information should the users provide**), processing( **analyzing of data**), and output ( **analysis report**) of the program
- ⌚ Justification of the choice of appropriate IT tools for solving the problem, such as **programming languages, compiler, platform, data structure**, etc)

### Design:

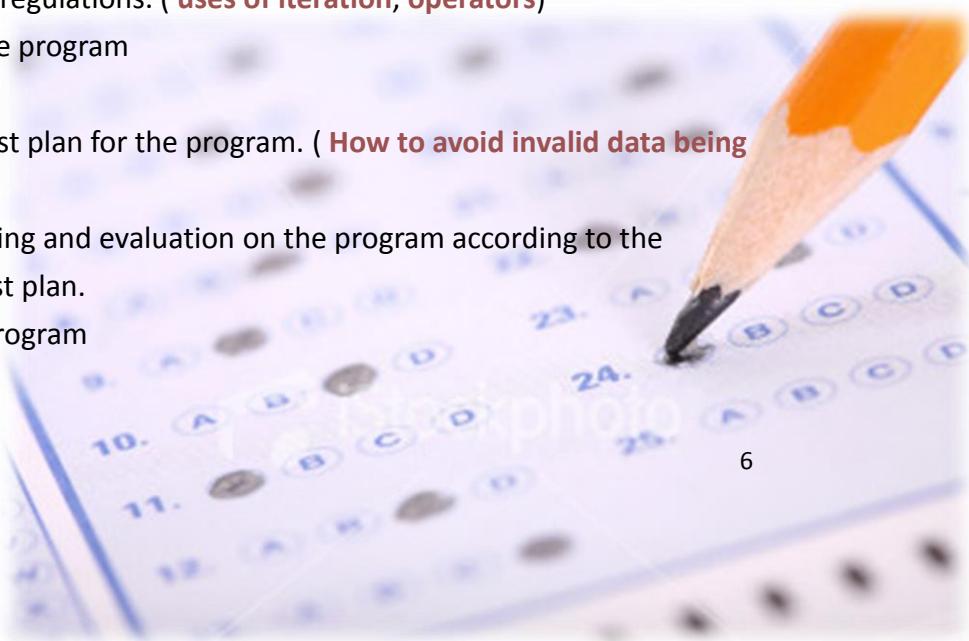
- ⌚ Design the overall structure of the program ( **main program, procedures, functions**, etc)
- ⌚ Design the **formats** of the **input files**
- ⌚ Design the **ways** that users should follow to **input data**
- ⌚ Design the **information** provided in the analysis report
- ⌚ Design the **layout** of the analysis report

### Implementation:

- ⌚ Decide the **data structures** that will be used in the program.
- ⌚ Design the **algorithms** according to the proposed functions, the competition regulations. ( **uses of iteration, operators**)
- ⌚ Construct the program

### Testing and Evaluation

- ⌚ Propose a test plan for the program. ( **How to avoid invalid data being processed?** )
- ⌚ Perform testing and evaluation on the program according to the proposed test plan.
- ⌚ Debug the program



## Multiple-Choice Analysis Report

### Conclusion and Discussion:

- ⌚ **Summary** of the project
- ⌚ **Conclusion** on what have been studied/learned
- ⌚ Discussion on the **favorable** features and **shortcomings** of the program
- ⌚ Suggestion on **further development/improvements** on the program.



# **Chapter 2 – Analysis**

## **2.1 Competition Regulations**

Before writing the program, the **regulations** of the competition should be clearly defined. For this, several **ICT competitions** existing in Hong Kong have been looked for, such as the Heat Event of the Hong Kong Olympiad in Informatics, which predominantly ask multiple choice questions. After careful considerations, regulations are defined as follows:

### **Participants:**

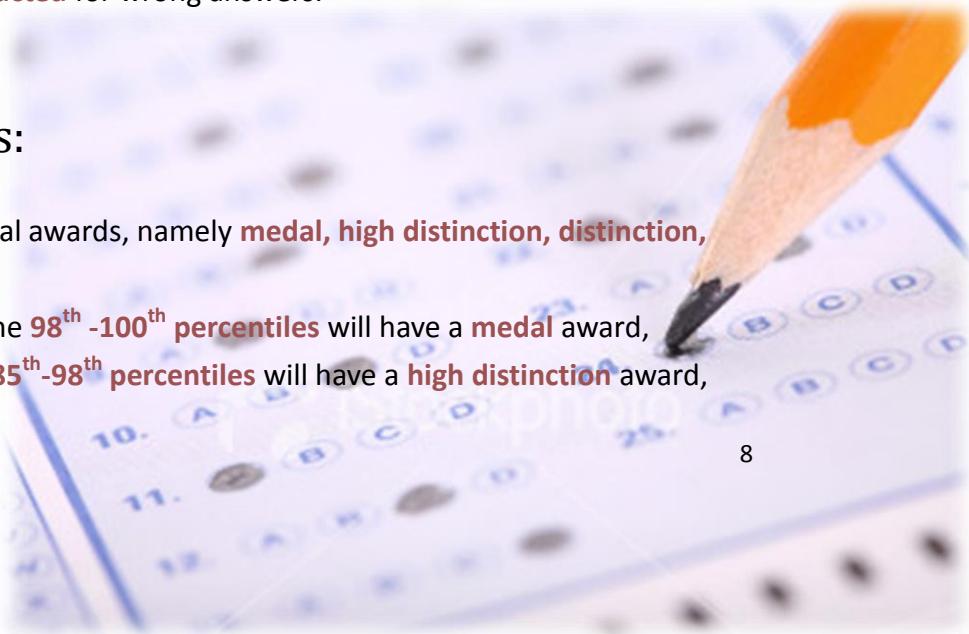
1. Only **school participants** are permitted to attend the competition.
2. All participants would attempt the **same set of questions** in the competition.
3. The number of participants is limited to **200**.

### **Questions:**

1. All questions are of **multiple choice type**.
2. There are **40** multiple choice questions in the competition.
3. There are **4** choices in each question.
4. The choices are **A,B,C and D**.
5. All questions are **compulsory** and they **carry equal marks**.
6. Candidates should only **mark one answer for each question**; marking more than one answer will score zero mark for that question.
7. Marks will **not be deducted** for wrong answers.

### **Individual awards:**

1. There are five individual awards, namely **medal, high distinction, distinction, merit and credit**.
2. Participant who is in the **98<sup>th</sup> -100<sup>th</sup> percentiles** will have a **medal** award,
3. the one who is in the **85<sup>th</sup>-98<sup>th</sup> percentiles** will have a **high distinction** award,



## Multiple-Choice Analysis Report

4. the one who is in the **70<sup>th</sup> -85<sup>th</sup> percentiles** will have a **distinction** award,
5. the one who is in the **50<sup>th</sup> -70<sup>th</sup> percentiles** will have a **merit** award
6. the one who is in the **35<sup>th</sup>-50<sup>th</sup> percentiles** will have a credit award.

### School awards:

1. Schools are ranked according to the **sum of the score** of all of the school participants.
2. There will be three awards, **champion award, 1<sup>st</sup> runner up award and 2<sup>nd</sup> runner up award.**
3. The school with the highest sum of score will have the champion award, the next will have the 1<sup>st</sup> runner up award and the next will get the 2<sup>nd</sup> runner up award.

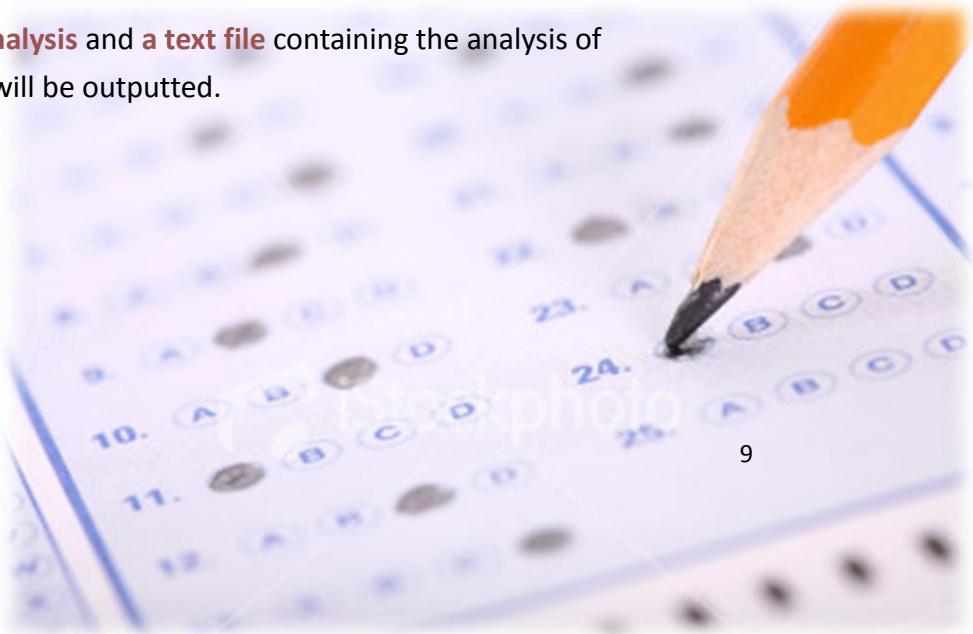
The program will therefore be written according to the above regulations.

## 2.2 Data collection

Data collection must be done before we have any further manipulations when we run the program. It is done by **reading the answer key and the answers of participants which are in text form**. The **format** and **details** of the text will be further discussed in 2.3 Input, Output and Process.

For convenience when scanning answers using OMR, a **single text file** containing all participants' answers and information will be used for storing the answers of participants. The program will read this file for **comparison** to the answer key and **do analysis** on participants' performance.

After analysis, **on-screen analysis** and **a text file** containing the analysis of participants' performance will be outputted.



## 2.3 Input, Output and Process

### Inputting data:

Users are required to enter the **name** of the text file of **MC answer key** and **participants' information and answers**, such that the program can be used for different answer sets with different file names as a MC analyzer for **not only one particular** ICT competition. Users can **alter** the answer key and participants' answers for **different analysis**. Also, it **reduces error** when users are trying to type in the answer **manually**.

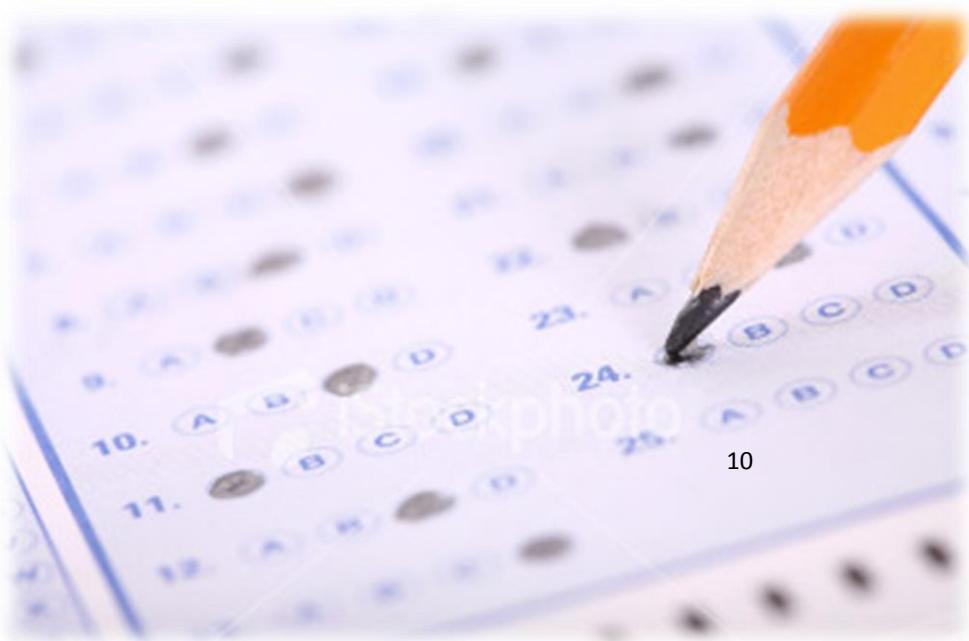
Users need to prepare **three text files** for the program, one containing the **answer key** of the questions and one containing the **participants' information and their answers**, lastly, one containing the **list of schools** that the participants are in.

### Format:

The MC answer key will be in a format of:

ABCDABCDABCABCDDBCABCDBABCDBABCDBCAABC

These answers will be read into the program and be compared with participants' answers.



## Outputting report:

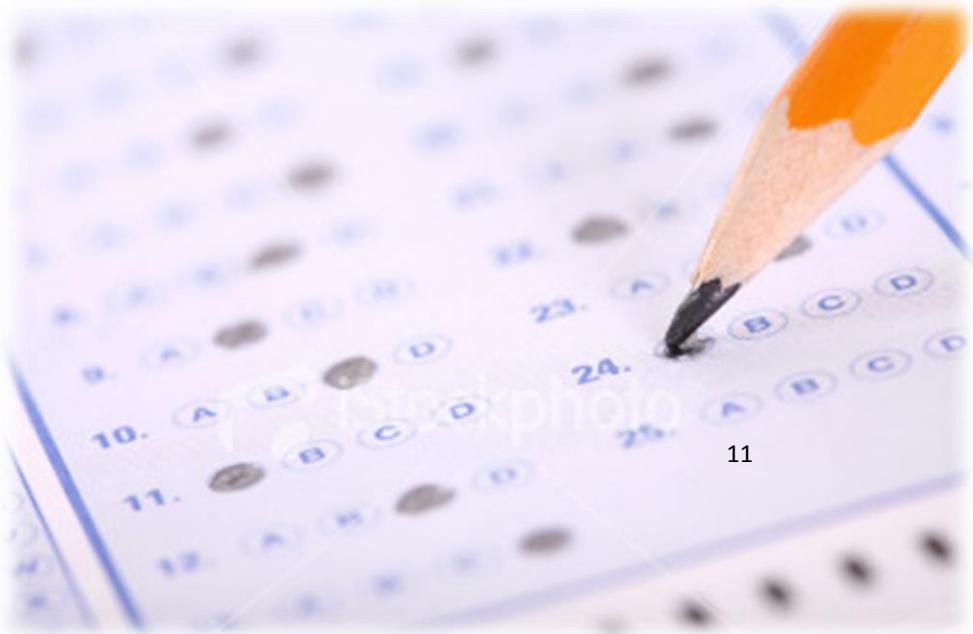
As stated in 1.2 Objectives, the program should be at least able to produce an analysis report having the followings:

- Total number of participants, total number of participating schools and total number of participant(s) from each participating school
- Winners of individual awards and school awards
- Question analysis, such as the percentage correct for each question

And I would like to add the followings:

- **Standard deviation** of score in the whole competition.
- **Median / mean** score of in the whole competition.
- **Range, inter-quartile range** of score
- **Upper and lower quartile** of score

The analysis will first be shown **on screen**, if requested by users, **a plain text form analysis report** can be generated, named by users. This makes the use of report more **flexible** and **user-friendly**.



## Data Processing

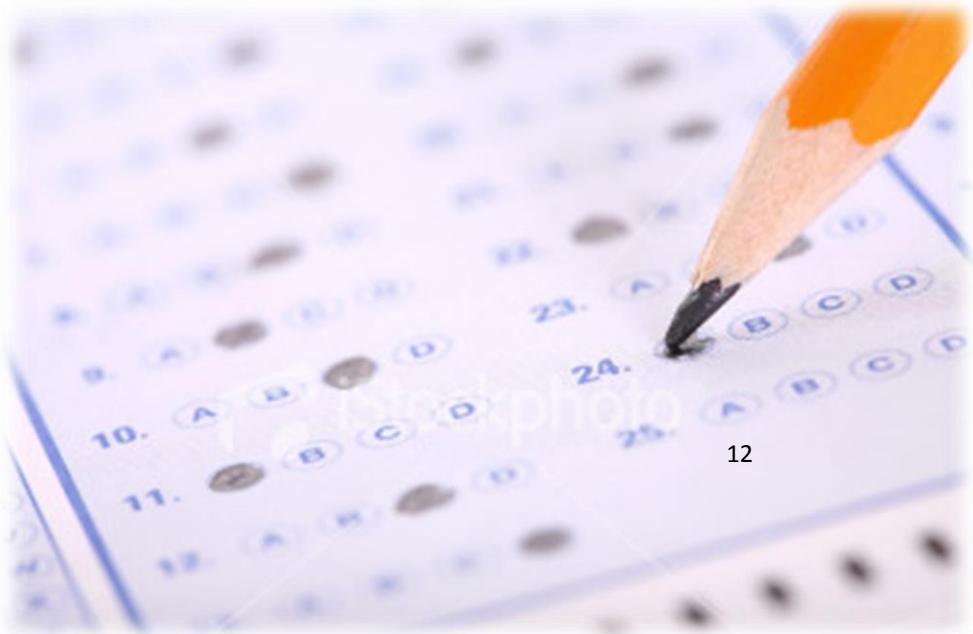
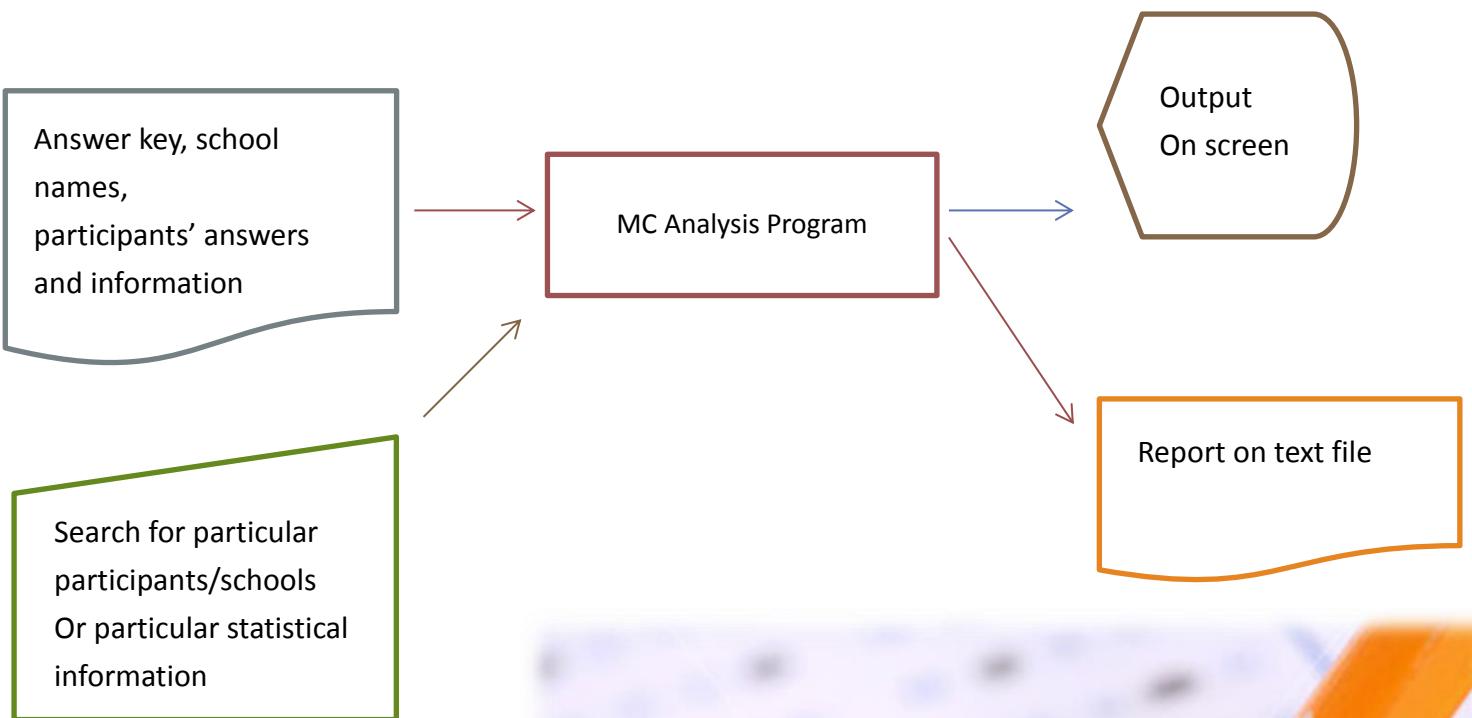
The **calculations on statistics**, sorting, searching and data file handling are involved in the program.

For example, we need to think of ways calculate the standard deviation of score using this formula

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

, which is quite **complicated** and **need efforts to fit it** into the program.

We also need to decide what **algorithms**, like bubble sort, to use, for ranking the results of schools to decide who will be awarded prizes. Also, we need to decide what searching algorithms to use, like binary search, to search for a particular record of a participant. Data file handling functions are also required to read and output text files.



## 2.4 Choice of IT Tools

### Choice of programming language

In order to choose the best programming language for writing the program, several **comparisons** have been made between different programming languages that are common nowadays.

Web-based programming language is not considered in this case because a web server is required. Also, I have no prior knowledge on web-based programming and it is hard to implement a web-based system.

#### Pascal

Pascal is a procedural language that is often used to teach programming due to its use of keywords instead of C-style braces and symbols, so it is a bit easier for beginners to understand than languages like C++.

#### C/C++

C is a popular language, especially in game programming, because it makes programs slightly faster and smaller than programs written in other languages, while C++ is well-suited for large projects because it has an object-oriented structure.

#### Java

Java is a multi-platform language that is especially useful in networking since it has java virtual machine which can be installed in various OS platforms such that programs of java can be easily exected

#### Visual Basic

Visual Basic is an event-driven language developed by Microsoft Corporation based on BASIC.

	Pros	Cons
Pascal	<p>1. I have been using this language for several years, so I am <b>quite familiar</b> with it.</p> <p>2. Its use of keywords makes reading, writing and editing the codes of this language <b>much easier because it is more English-like</b>, compared to other languages.</p>	<p>1. It is <b>hard to produce pictures</b>, animations and dynamic interface.</p> <p>2. The use of command-line interface makes inputting <b>not so user-friendly</b>.</p>
C/C++	<p>1. Its <b>processing speed is fast</b> compared to other listed programming languages.</p> <p>2. C++ provides a useful STL (Standard Template Library) which can <b>save up time typing many codes</b>.</p>	<p>1. I have <b>no prior knowledge</b> about C programming.</p> <p>2. Its <b>syntax is quite complicated</b> and will cost a lot of time to learn.</p>
Java	<p>1. Java program are <b>platform independent</b>, it can be run on any PC or any operation system.</p> <p>2. It is purely OOPS language having all the OOPS, concept, <b>which saves up time to code</b>.</p>	<p>1. I have <b>no prior knowledge</b> about Java programming.</p> <p>2. It <b>takes more time to execute</b> than e.g. C language as it is run first on JVM (Java Virtual Machine)</p> <p>3. <b>More memory consuming</b> than other languages.</p>
Visual Basic	<p>1. It has a <b>graphic user interface (GUI)</b> that can be used by beginners with ease. Programs can be created by using the drag and drop feature that doesn't require a person to write any coding.</p>	<p>1. I have <b>no prior knowledge</b> about Visual Basic programming.</p> <p>2. Visual basic has <b>low flexibility</b> as it is a proprietary programming language written by Microsoft, so programs written in Visual basic cannot, easily, be transferred to other operating systems.</p>

## Multiple-Choice Analysis Report

Comparing the pros and cons of each programming language, I decided to use **Pascal** as the programming language to write the program.

Firstly, I have **prior knowledge about Pascal programming** that I could write programming codes and use different operations in Pascal at ease.

Secondly, its use of keywords makes **reading, writing and editing the codes of this language much easier because it is more English-like**, compared to other languages, which is very critical when it comes to testing and debugging at the final stage of this system.

Also, the **functions** provided by Pascal are **good enough** for me, for example, Pascal can do text manipulation easily and so I can accomplish the tasks easily too. Dynamic interface or GUI is not necessarily needed in this case because the ultimate goal of the program is to generate a MC analysis report.

Using Pascal to finish the program can also **refine my programming techniques**, which are important in the curriculum of ICT in HKDSE, as I could **learn and revise Pascal programming in a practical way**.



## Choice of Pascal compiler

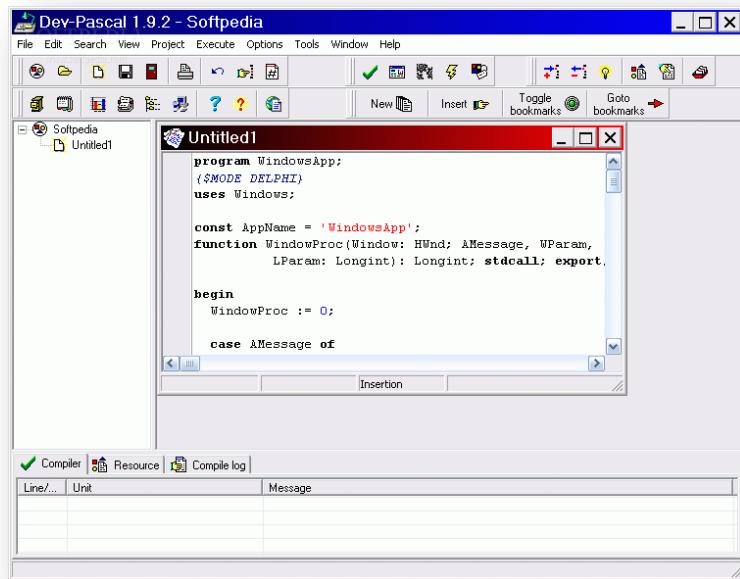
There are 4 common Pascal compilers that I can use. They are Quick Pascal, Turbo Pascal , Free Pascal and Dev-Pascal.

	<b>Pros</b>	<b>Cons</b>
<b>Quick Pascal</b>	/	<ul style="list-style-type: none"> <li>● It does not support a <b>large array</b></li> <li>● Programmers <b>can only use keyboard to code</b>,</li> <li>copying/pasting using a mouse is not possible.</li> </ul>
<b>Turbo Pascal</b>	/	<ul style="list-style-type: none"> <li>● It does not support a <b>large array</b></li> </ul>
<b>Free Pascal</b>	<ul style="list-style-type: none"> <li>● Programmers can <b>use a mouse to assist coding</b>; copying/pasting using a mouse is possible.</li> </ul>	<ul style="list-style-type: none"> <li>● The <b>compile time of Free Pascal is slower</b> than that of Dev-Pascal.</li> </ul>
<b>Dev-Pascal</b>	<ul style="list-style-type: none"> <li>● Programmers <b>can use a mouse to assist coding</b>, copying/pasting using a mouse is possible.</li> <li>● I have <b>always been using Dev-Pascal</b> to compile a program. It is more familiar to me.</li> <li>● It supports a <b>large array</b>.</li> <li>● It has a <b>built-in program library</b>. Many functions are available to use.</li> <li>● It <b>supports various OS platforms</b>.</li> </ul>	/

## Multiple-Choice Analysis Report

From the above table, it is clearly seen that **Dev-Pascal** is a better compiler than others, for its **useful built-in functions** and **ease of coding**. So I will choose Dev-Pascal as the compiler.

### Dev-Pascal interface

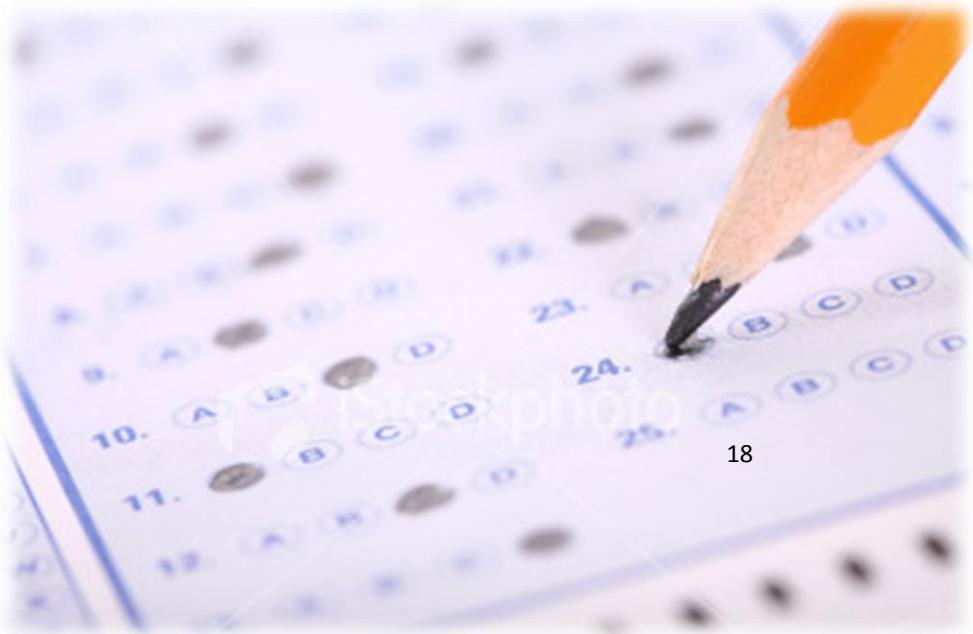
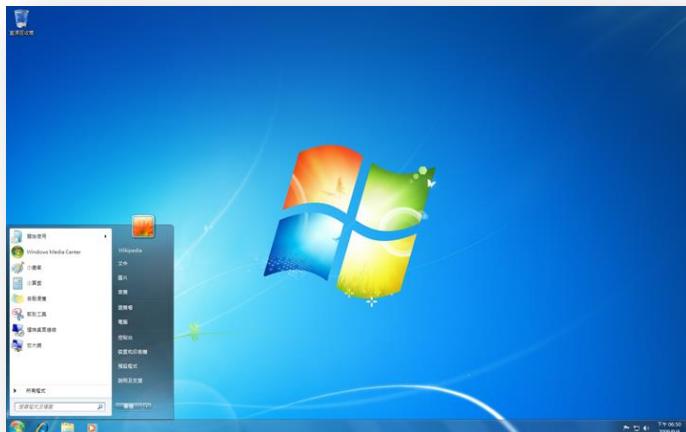


## Choice of OS platform

To be **compatible with most users**, the program should be written to be used on the **OS platform used by most people**. From **w3schools**, a web developer information website launched from 1999, until December 2013, the percentages of users of the major OS systems are as follows:

2013	Win8	Win7	WinXP	Linux
December	8.2%	55.9%	11.6%	4.8%

**Window 7** and **Window XP** occupy the most percentages in the market, which shows that they are the most popular OS systems in the market. With the fact that I being a both Window 7 and Window XP user, it is easy for me to test for and write the program on these two platforms. I may also invite external user to test for me the compatibility of the program on **Window 8** too.



## Choice of hardware to solve the problem

Below is a list of hardware that a user needs to run the program:

**Personal computer(PC):** it should have at least a hard disk to store and read data of the MC analysis program and ports to connect the keyboard and the VDU

**Keyboard:** to give commands to the program

**Visual display unit (VDU):** a monitor to display the program and show the analysis report

## Choice of other software to solve the problem

Below is a list of software that is needed to solve the problem:

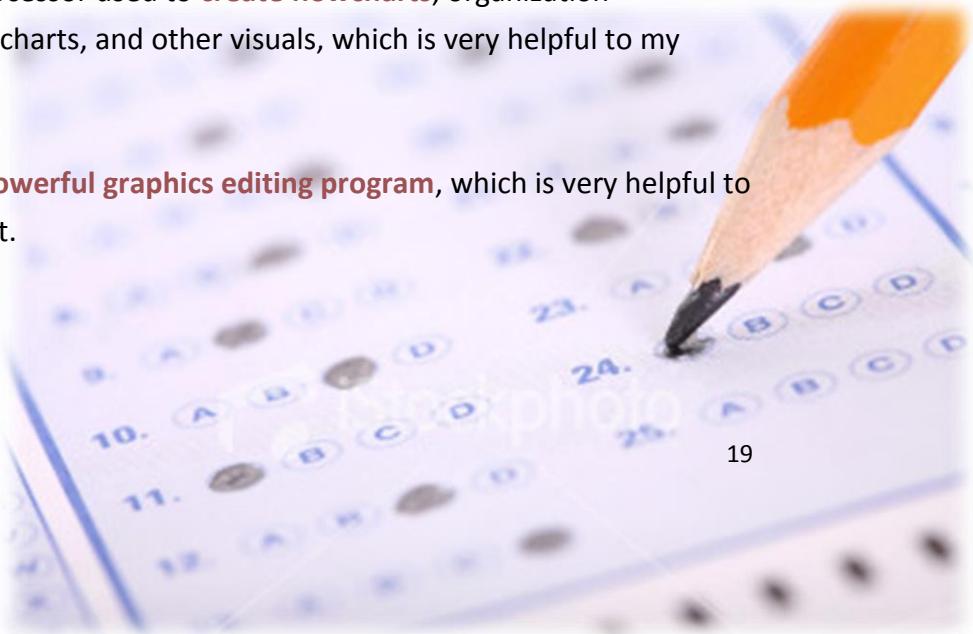
**Notepad:** for **preparing data file** (answer keys, participants' information and answers)

**Google Chrome (web browser) :** for **searching information** which helps developing the program

**Microsoft Word 2010:** it provides **word processing** function that I can write my report for this project and design the layout and content of the analysis report well. Also, the graphics and graphics editing functions provided in the 2010 version are good enough for me to draw some simple charts.

**SmartDraw:** it is a visual processor used to **create flowcharts**, organization charts, mind maps, project charts, and other visuals, which is very helpful to my illustration in this report.

**Adobe Photoshop:** it is a **powerful graphics editing program**, which is very helpful to my illustration in this report.



## Assumptions

The following assumptions have been made about the users:

- They have a computer running Window 7 or Window XP which has a keyboard and VDU.
- They are capable of typing commands through a keyboard.
- They are capable of reading English prompt words.
- The ICT competition is held according to the regulations stated in 2.1 Competition Regulations
- There are non-limited number of schools participated in the competition.
- There are 200 participants in the primary testing of the program.



# **Chapter 3- Design of Solution**

## **3.1 Brief Description**

In this Chapter, I will design the program based on the functions I proposed in Chapter 1, and the input, process and output I studied in Chapter 2.

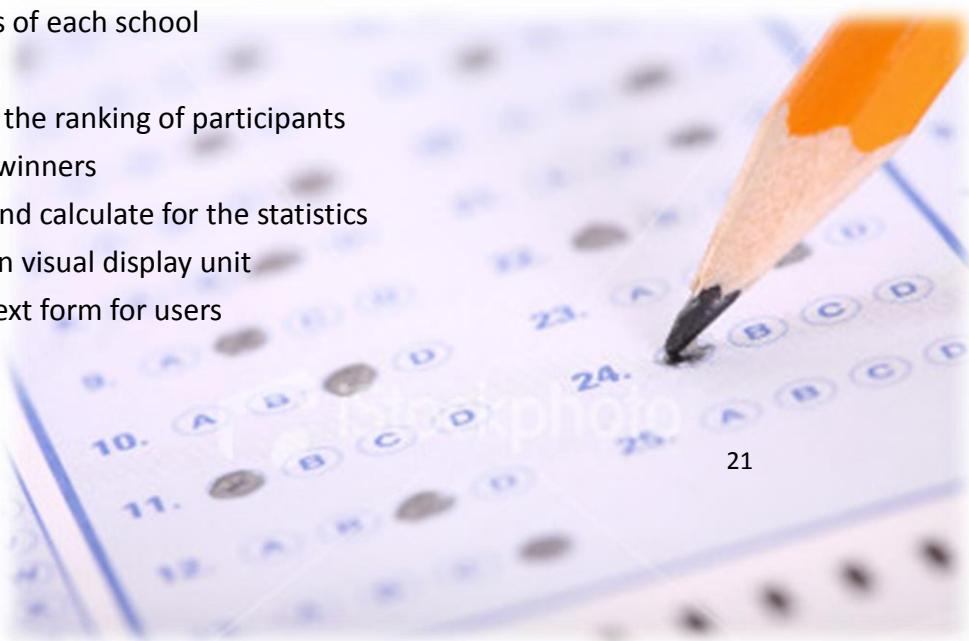
I will design:

- ⌚ the **overall structure** of the program by refining the problem
- ⌚ the **formats** of the data files for storing the answers of the participants and the answer keys
- ⌚ **prompt words** that navigate users to control the program
- ⌚ the **screen layout** of the main menu and sub-menu for users to control the program
- ⌚ the **layout and content** of the report and its text file for storing the analysis report

## **3.2 Refinement of Problem**

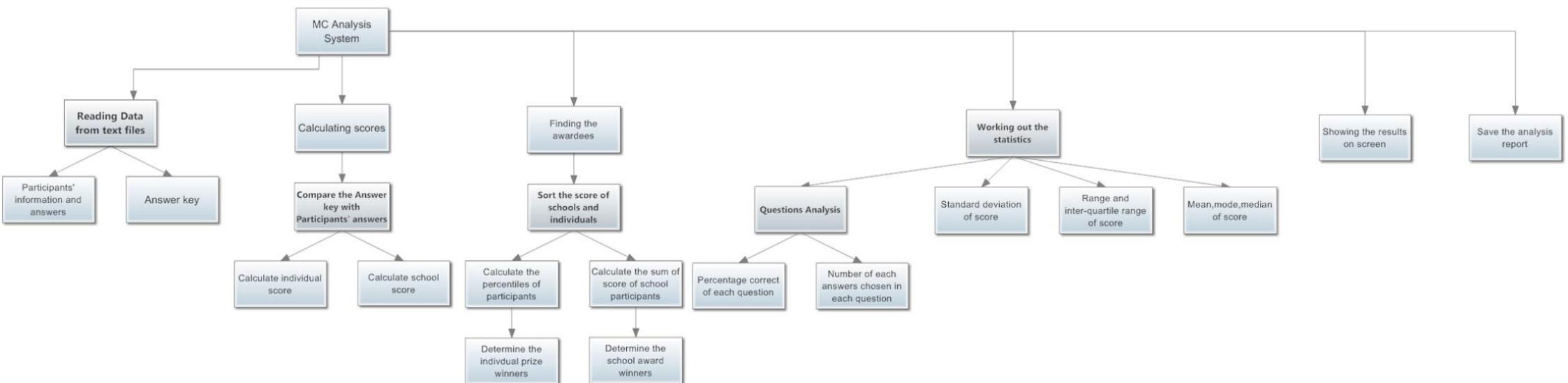
As stated in Chapter 1, the problem my program is going to solve can be divided into several sub-problems, they are:

- ⌚ To read the answer and the information of participants
- ⌚ To read the answer key for the questions
- ⌚ To compare the answers of participants to the answer key
- ⌚ To count the number of participants
- ⌚ To calculate the score of each participant
- ⌚ To calculate the results of each school
- ⌚ To store these data
- ⌚ To sort the records for the ranking of participants
- ⌚ To find out the award winners
- ⌚ To analyze questions and calculate for the statistics
- ⌚ To generate a report on visual display unit
- ⌚ To save the report in text form for users



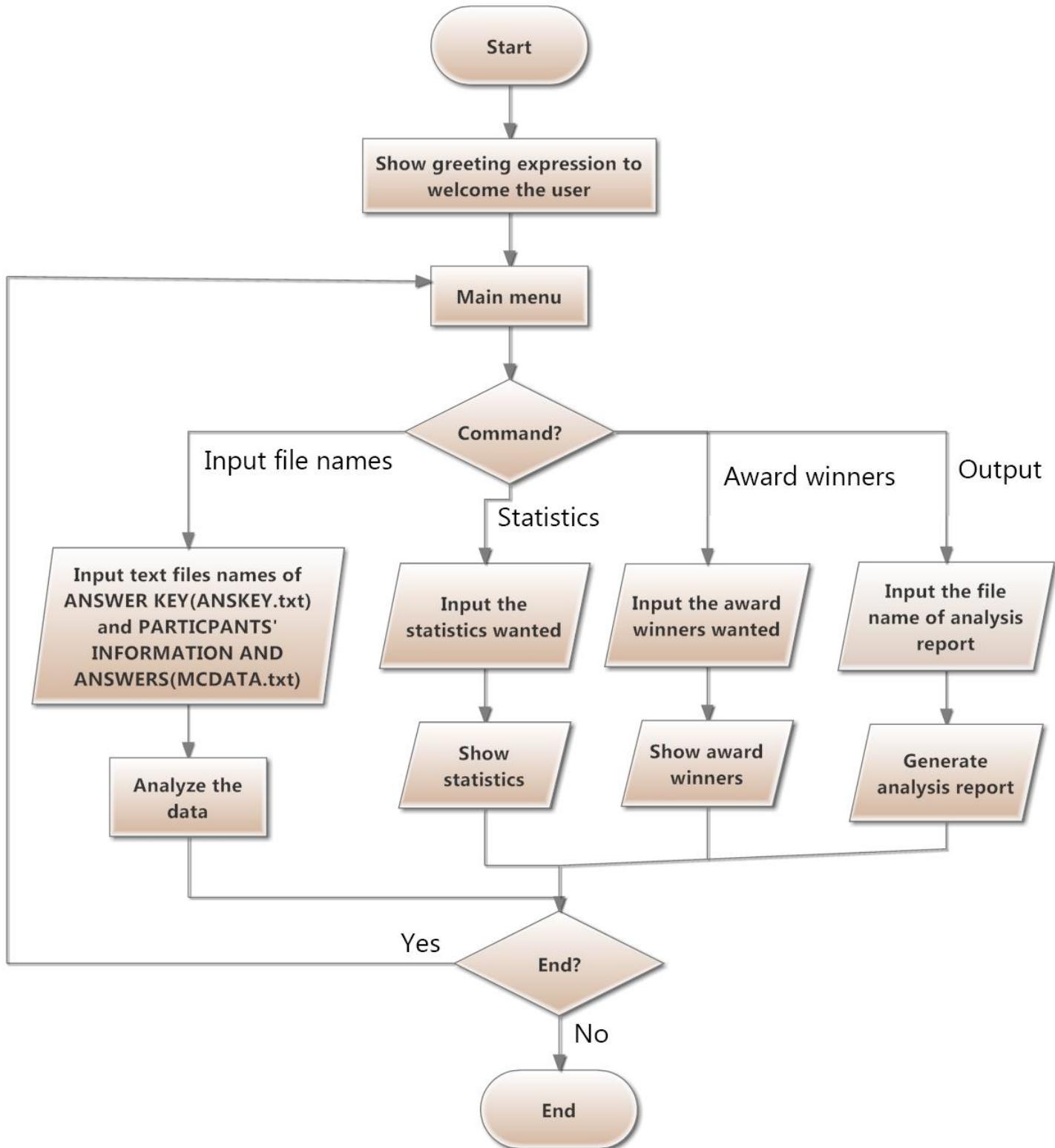
## Multiple-Choice Analysis Report

And these sub-problems can be represented by a structure chart:



## Multiple-Choice Analysis Report

To further illustrate the flow of the program clearly, a flowchart of the program can be shown as follows:



As greeting the user will be a **friendly** way to welcome the user to use the program, a greeting screen will be designed also. This will be further discussed in the coming sections.

## 3.2 Input data method and file formats

### Input data method

For users to input to control the program, an input data method has to be decided. For Pascal program, users usually use **keyboard** to input. The most convenient way to do that is **typing in the corresponding command number** to control the program.

```
COMPETITION MC ANALYSIS
*****
Numbers of participations
*****
1. Total number of participants
2. Total number of schools participated
3. Total number of participants from each schools

Questions and Score Analysis
*****
6. Percentage correct for each questions
7. Standard deviation
8. Mean/ Median /Mode
9. Range/ Upper.lower-quartile/ Inter-quartile range

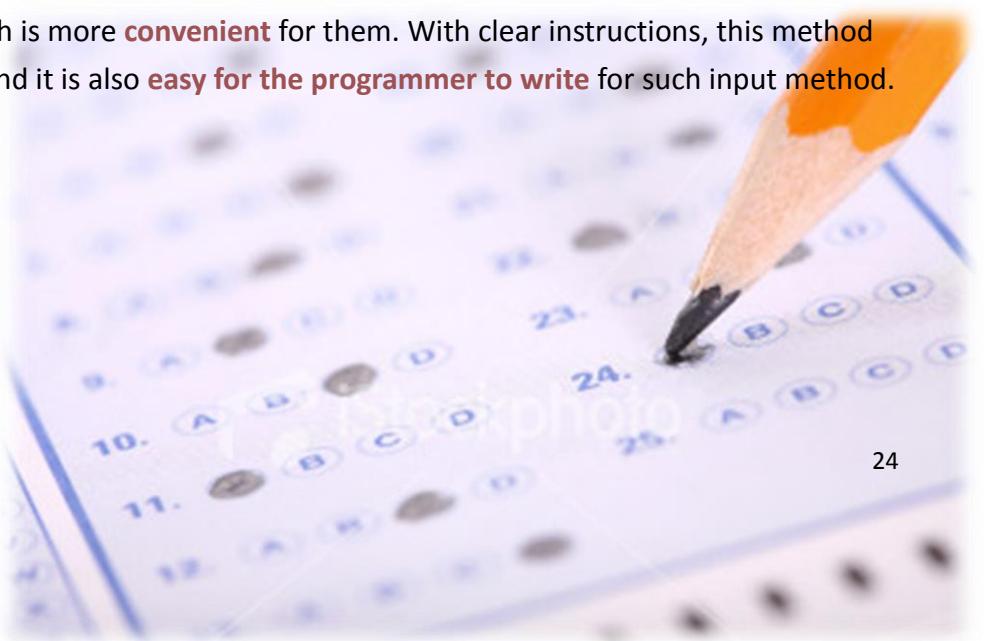
Awardees
*****
4. Individual awardees
5. School awardees

Search
*****
10. By name
11. By school
12. By sex

13. Display the list of all participants
14. Save report in text form
15. Quit

Please enter the option<1-15>: 2
```

The users can know **what to type in easily** by reading the prompt words. Also, the **colours** of the prompt words and the words inputted by the user will be **set to be different** so that the users **will not mix them up**, which is more **convenient** for them. With clear instructions, this method is therefore **user-friendly** and it is also **easy for the programmer to write** for such input method.



## File formats

### File storing the Participants' Answers and information:

It consists of **two files**, one storing their **answers and basic information** (e.g. sex), and one storing **the list of schools** that the participants are in.

A sample file name of it is assigned to be MCDATA.TXT and SCHOOL.TXT .The program will **accept any file names** other than that but the user **must make sure the file follows the format** discussed in this section. This is to **avoid errors to occur while the program is working out the statistics**, in case other materials are read instead of the file.

The file type of it will be **plain text file**, because **all personal computers can write or read text file**, and it is **easy for the programmer to write** procedures containing plain text file manipulation.

The file stores the record of each participant, which includes the following information:

- ⌚ **Name** of the participant ( 30 characters)
- ⌚ **Sex** of the participant (1 character Male(M)/Female(F) )
- ⌚ **School name** of the participant ( 50 characters)
- ⌚ **Answers** of the 40 questions (altogether 40 characters)
  - Valid answers: 'A', 'B', 'C', 'D'
  - Invalid / Null/More than one answers: '-'

Each line of the file stores the record of one participant with the following format:

Name of the participant ( 25 characters)	Sex (1 character)	School name ( 45 characters)	Answers 40 characters in 'A', 'B', 'C', 'D', '-'
e.g. Mo Cheuk Yin	M	Cheung Sha Wan Catholic Secondary School	A    B    C.....    D    A



## Multiple-Choice Analysis Report

The **number of characters** of the **names** of participants and **school names** of participants are assigned to be **25** and **45** respectively because those are the numbers **good enough** to **store** most of the **names of people** and **schools** in **Hong Kong**, including some **foreign names**.

Sample file:

MCDATA.txt - 記事本		
檔案(F)	編輯(E)	格式(O)
檢視(V)	說明(H)	
Au Wing Hing	M Diocesan Boy's School	ADECADBEBCDCCACDDEDBDCEDDCCCADDCCABBDEAB
Wong Tin Long	M Ying Wa College	DDCBBBBDACADCCAAECBBCDEEDDAACCACABEDADEBD
Chan Tai Man	M Wah Yan College, Kowloon	BEEAACDCBDABDDEADCBCCCCCCACEADABBCBCACDAC
Wong Sai Man	M Wah Yan College, Kowloon	ABCBCBBADADCABDCABDCAACDBADBADACBCBACBCDAAA
Chan Tin Wai	M Diocesan Boy's School	ADACDBACDBADBDCDCABDDBADACBCDADCBDCDADCA
Wong Tung Tung	M Venus College	CBCBADACDCADCABDCDADCBACDBABDADACBADA
Chan Chun Hung	M Earth College	ADACDBACDBADBDCDCABDDBADACBCDADCBDCDADCA
Law Chi Ho	M Mars College	BEEAACDCBDABDDEADCBCCCCCCACEADABBCBCACDAC
Wong Ka Lok	M League School	ADACDBACDBADBDCDCABDDBADACBCDADCBDCDADCA
Chan Ka Kin	M Maple Tree College	BEEAACDCBDABDDEADCBCCCCCCACEADABBCBCACDAC

The file storing the list of schools will only contain the **names of school**. The **maximum length** for each name is again, **45** characters.

Sample file:

SCHOOL.txt - 記事本		
檔案(F)	編輯(E)	格式(O)
檢視(V)	說明(H)	
FALL SECONDARY SCHOOL		
FIRE SCHOOL		
ATMOSPHERE COLLEGE		
SPRING SCHOOL		
SUMMER SCHOOL		
WINTER SECONDARY SCHOOL		
VENUS COLLEGE		
EARTH COLLEGE		
MARS COLLEGE		
LEAGUE SCHOOL		
MAPLE TREE COLLEGE		
HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY		
SUN MEMORIAL SCHOOL		
MOON SECONDARY SCHOOL		

## File storing the answer key:

It consists of **one single file** only.

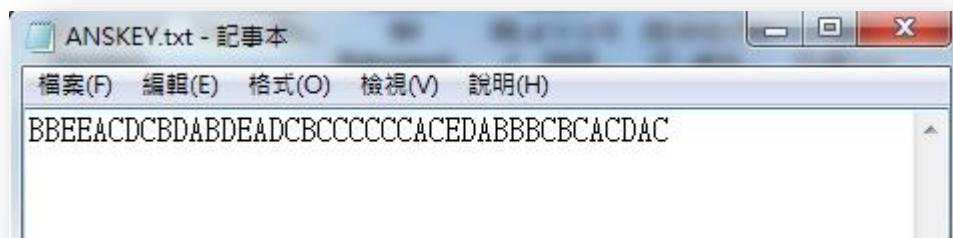
A sample file name of it is assigned to be ANSKEY.TXT. The program will **accept any file names** other than that but the user **must make sure the file follows the format** discussed in this section. Again, this is to **avoid errors to occur while the program is working out the statistics**, in case other materials are read instead of the file.

## Multiple-Choice Analysis Report

The file type of it will be **plain text file**, because **all personal computers can write or read text file**, and it is **easy for the programmer to write** procedures containing plain text file manipulation.

The file stores **one set** of **answer keys** for the 40 MC questions. The answer keys consist of 40 characters in 'A', 'B', 'C', 'D'.

For example,



Sample input screen for inputting file name on running the program

```
8 8888 ,o8888888o.8888888 8888888888
8 8888 8888 `88. 8 8888
8 8888 ,8 8888 `8. 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 `8 8888 .8 8 8888
8 8888 8888 ,88 8 8888
8 8888 `8888888P 8 8888      COMPETITION MC ANALYSIS
*****
Please input the file name of the participants' information and answers
default:<MCADATA.txt> :MCADATA.txt
```

The user can either input the name of file **by keyboard** like this or, the user may input the file name by **drag and drop to give the full file path**:

```
8 8888 ,o8888888o.8888888 8888888888
8 8888 8888 `88. 8 8888
8 8888 ,8 8888 `8. 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 `8 8888 .8 8 8888
8 8888 8888 ,88 8 8888
8 8888 `8888888P 8 8888      COMPETITION MC ANALYSIS
*****
Please input the file name of the participants' information and answers
default:<MCADATA.txt> :F:\MCANALYSIS\MCDATA.txt
```

## 3.3 Output Format

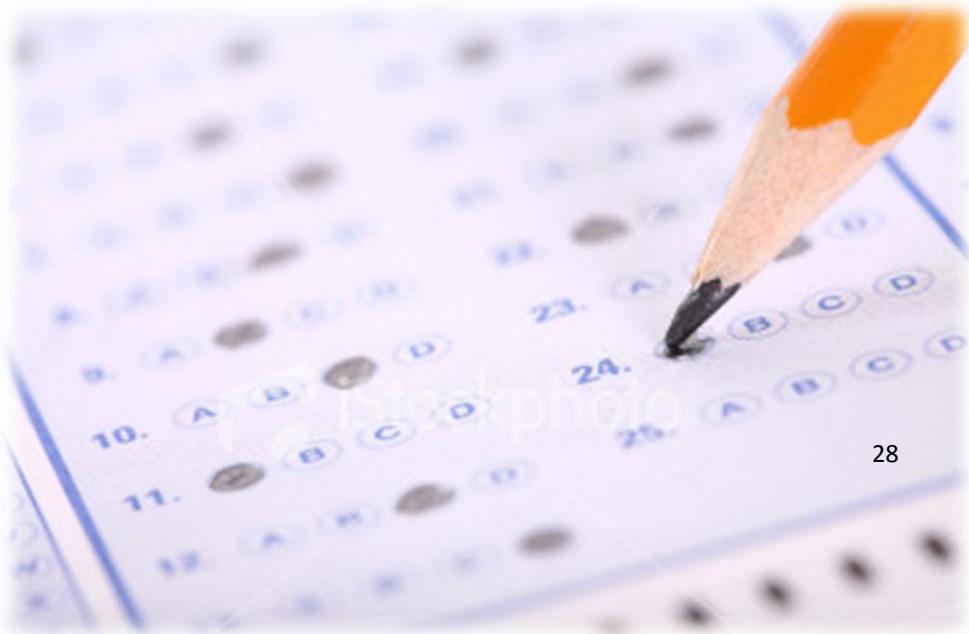
There are **two kinds** of output format in this program: **screen output** and **text file**.

### Screen output

In screen output, only the **information requested** by the user will be **displayed** as the display on screen is of **limited size**, not too much information can be displayed on one screen.

The following information can be requested by the user to see:

- Total number of participants, total number of participating schools and total number of participant(s) from each participating school
- Winners of individual awards and school awards
- Percentage correct for each question
- Standard deviation of score in the whole competition.
- Median / mode/mean score of in the whole competition.
- Range, inter-quartile range of score
- Score in upper quartile and lower quartile



# Multiple-Choice Analysis Report

For example, for listing a list of participants, the **whole page of a default screen** will be used.

Total number of participants from each schools	
School	No. of participants
SUMMER SCHOOL	1
FIRE SCHOOL	1
ATMOSPHERE COLLEGE	1
LEAGUE SCHOOL	15
MOON SECONDARY SCHOOL	14
EARTH COLLEGE	15
MAPLE TREE COLLEGE	15
SPRING SCHOOL	15
MARS COLLEGE	18
FALL SECONDARY SCHOOL	16
VENUS COLLEGE	19
SUN MEMORIAL SCHOOL	23
WINTER SECONDARY SCHOOL	22
HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	25

The above screen shows an example of showing the total number of participants from each school, the names of school and their corresponding number of participants is clearly displayed.

For small amount of information being displayed, only **a few lines** on the screen will be used to display the information required from the users, for example,

**Numbers of participations**

\*\*\*\*\*

1. Total number of participants  
2. Total number of schools participated  
3. Total number of participants from each schools

**Questions and Score Analysis**

\*\*\*\*\*

6. Percentage correct for each questions  
7. Standard deviation  
8. Mean/ Median /Mode  
9. Range/ Upper,lower-quartile/ Inter-quartile range

13. Display the list of all participants  
14. Save report in text form  
15. Quit

Please enter the option<1-15>: 1

Total number of participants for the competition: 200  
Total number of male participants: 118  
Total number of female participants: 82  
Press <enter> to continue

## Multiple-Choice Analysis Report

The above screen shot shows an example when the user enquires “total number of participants” from the program, just a few lines are displayed.

## Output results in analysis report file

It consists of **one single text file** only.

As discussed in [2.3 Input, Output and Process](#), the file name will be **inputted by the user** for flexibility.

The file type of the report will be **plain text file** for its **ease** for Pascal program to generate.

All the results that can be requested in the screen output section will be shown in the report.( Please refer to [page 21](#).)

A sample layout of analysis report:

A Report on the ICT competition

-----  
Participation:

\*\*\*\*\*

Total number of participants for the competition: 200

Total number of male participants: 118

Total number of female participants: 82

Total number of schools for the competition: 14

Total number of participants from each schools

\*\*\*\*\*

School

No. of participants

\*\*\*\*\*

\*\*

SUMMER SCHOOL

1

FIRE SCHOOL

1

ATMOSPHERE COLLEGE

1

LEAGUE SCHOOL

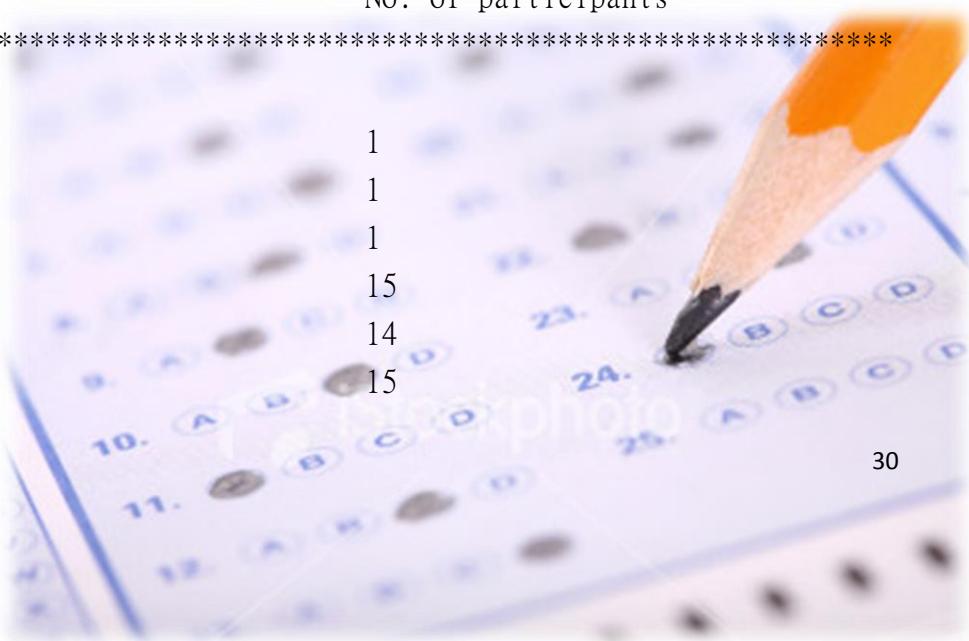
15

MOON SECONDARY SCHOOL

14

EARTH COLLEGE

15



## Multiple-Choice Analysis Report

MAPLE TREE COLLEGE	15
SPRING SCHOOL	15
MARS COLLEGE	18
FALL SECONDARY SCHOOL	16
VENUS COLLEGE	19
SUN MEMORIAL SCHOOL	23
WINTER SECONDARY SCHOOL	22
HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	25

### AWARDS:

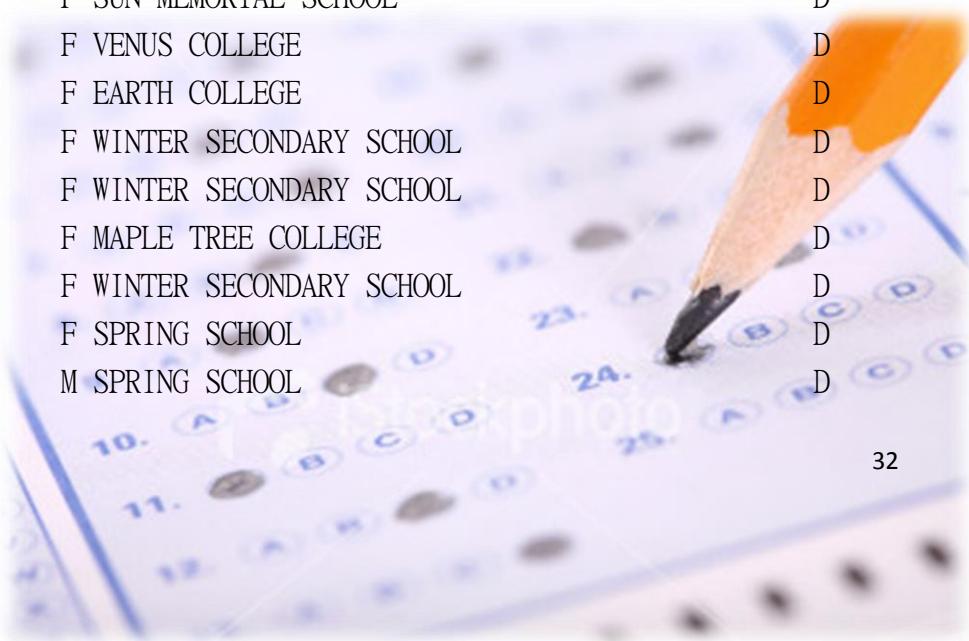
Individual awardees of the competition

\*\*\*\*\*

Name	Sex	School	Award
**			
LI KA SHING	M	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Medal
NG SIU YEE	F	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Medal
LEUNG HIU FUNG	M	SPRING SCHOOL	Medal
HOWARD YOUNG	M	FALL SECONDARY SCHOOL	Medal
SO PANG HIN	M	MOON SECONDARY SCHOOL	Medal
ADAM HENDERSON	M	VENUS COLLEGE	HD
CHAN TIN WAI	M	WINTER SECONDARY SCHOOL	HD
FONG CHEUK HO	M	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	HD
CHAN PUI HAN	F	MOON SECONDARY SCHOOL	HD
CYNTHIA MARTIN	F	VENUS COLLEGE	HD
BRENDA RIVERA	F	MARS COLLEGE	HD
JANICE BROOKS	F	VENUS COLLEGE	HD
TAM WING YAN	F	FALL SECONDARY SCHOOL	HD
CARL ALEXANDER	M	MAPLE TREE COLLEGE	HD
TERESA JENKINS	F	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	HD
WONG NGA YIN BELINDA	F	EARTH COLLEGE	HD
LEE SHU JEN	M	LEAGUE SCHOOL	HD
LEUNG WAI CHI	M	EARTH COLLEGE	HD
LUI KEI KWAN	M	LEAGUE SCHOOL	HD
RAYMOND COX	M	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	HD
SENA KOBAYAKAWA	M	MAPLE TREE COLLEGE	HD

## Multiple-Choice Analysis Report

MARILYN RUSSELL	F VENUS COLLEGE	HD
KATHY BAKER	F SUN MEMORIAL SCHOOL	HD
CHAN WING YEE	F SPRING SCHOOL	HD
AU WING HING	M FALL SECONDARY SCHOOL	HD
CHU KIN FUNG	M WINTER SECONDARY SCHOOL	HD
FONG KA WING	M FALL SECONDARY SCHOOL	HD
LAM MAN CHUNG	M MAPLE TREE COLLEGE	HD
LAW HO YU	M FALL SECONDARY SCHOOL	HD
MAR TIN PONG	M SUN MEMORIAL SCHOOL	HD
NG WAI CHUNG	M WINTER SECONDARY SCHOOL	HD
GARY WOOD	M MAPLE TREE COLLEGE	D
TAYLOR WASHINGTON	M WINTER SECONDARY SCHOOL	D
GREGORY LONG	M MARS COLLEGE	D
LEUNG HOI YAN	F VENUS COLLEGE	D
DEBRA COX	F FALL SECONDARY SCHOOL	D
RUBY LOPEZ	F MOON SECONDARY SCHOOL	D
VIRGINIA PATTERSON	F SUN MEMORIAL SCHOOL	D
ANN WATSON	F WINTER SECONDARY SCHOOL	D
TONG YIU YAN	F EARTH COLLEGE	D
YU JIN REN	M ATMOSPHERE COLLEGE	D
HO HOI TO	M WINTER SECONDARY SCHOOL	D
CHAN HEUNG WAI	M VENUS COLLEGE	D
AU CHEUK KEI	M FALL SECONDARY SCHOOL	D
TSANG MAU YIU	M VENUS COLLEGE	D
ERIC COOPER	M MARS COLLEGE	D
PHILIP SMITH	M SUN MEMORIAL SCHOOL	D
JOSHUA THOMPSON	M MOON SECONDARY SCHOOL	D
BRUCE CARTER	M SUN MEMORIAL SCHOOL	D
DAIKICHI KOMUSUBI	M EARTH COLLEGE	D
CHEUNG HOI YAN	F SUN MEMORIAL SCHOOL	D
NORMA BELL	F VENUS COLLEGE	D
LISA WRIGHT	F EARTH COLLEGE	D
MARGARET BENNETT	F WINTER SECONDARY SCHOOL	D
KATHLEEN PEREZ	F WINTER SECONDARY SCHOOL	D
CHING TSAN WING	F MAPLE TREE COLLEGE	D
CHAN KA YI	F WINTER SECONDARY SCHOOL	D
YEUNG CHING YIU	F SPRING SCHOOL	D
CHAN TAI MAN	M SPRING SCHOOL	D



## Multiple-Choice Analysis Report

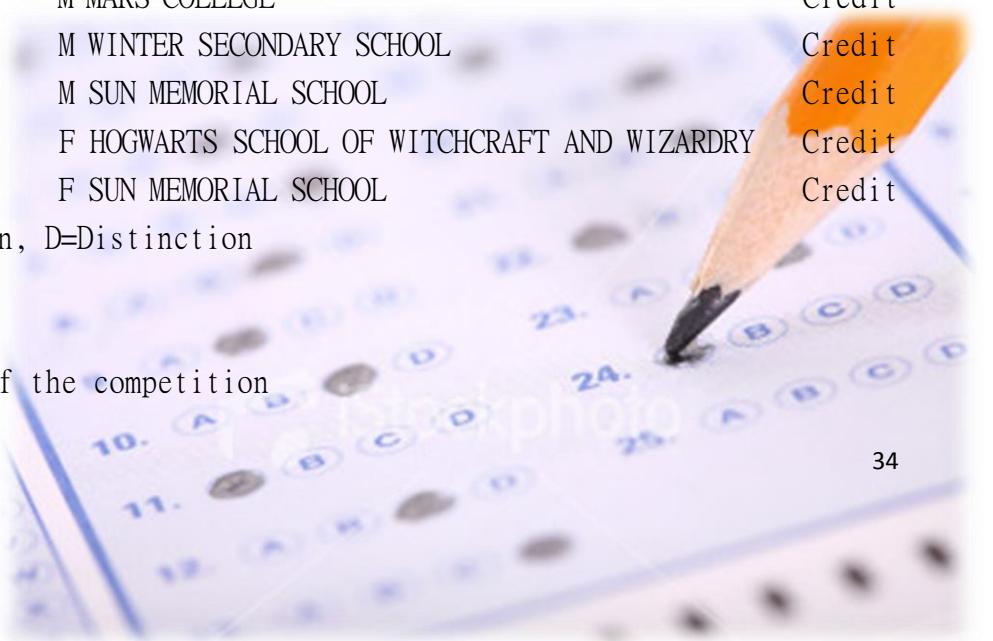
WONG KA LOK	M LEAGUE SCHOOL	D
CHAN SAU TIN JUSTIN	M EARTH COLLEGE	D
CHEUNG CHUN TING	M SUN MEMORIAL SCHOOL	Merit
CHEUNG WAI KEI	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Merit
LAI KA WING	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Merit
TO KENNETH	M SPRING SCHOOL	Merit
YIP KIM LUNG	M WINTER SECONDARY SCHOOL	Merit
CHAN PAK YIN	M MARS COLLEGE	Merit
LAM KIN CHUN	M EARTH COLLEGE	Merit
WONG SZE HO STANLEY	M SPRING SCHOOL	Merit
EUGENE LEE	M MOON SECONDARY SCHOOL	Merit
BOBBY PARKER	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Merit
JESSE BAILEY	M WINTER SECONDARY SCHOOL	Merit
AMPERE HUGHES	M MOON SECONDARY SCHOOL	Merit
JOSE CLARK	M VENUS COLLEGE	Merit
DAVID FLORES	M VENUS COLLEGE	Merit
JOE KING	M MOON SECONDARY SCHOOL	Merit
WONG SIU YAN	F SPRING SCHOOL	Merit
NG YEE KIU	F MAPLE TREE COLLEGE	Merit
MARTHA MARTINEZ	F WINTER SECONDARY SCHOOL	Merit
NANCY DIAZ	F MAPLE TREE COLLEGE	Merit
DEBORAH GRAY	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Merit
NICOLE JACKSON	F FALL SECONDARY SCHOOL	Merit
PHYLLIS LEWIS	F MAPLE TREE COLLEGE	Merit
CHAN KA KIN	M MAPLE TREE COLLEGE	Merit
MICHAEL JACKSON	M MOON SECONDARY SCHOOL	Merit
LAI KA WING	M EARTH COLLEGE	Merit
MOK SIU HEI	M FALL SECONDARY SCHOOL	Merit
WONG KA LOK	M FALL SECONDARY SCHOOL	Merit
CHU WAI HON	M MAPLE TREE COLLEGE	Merit
LUI KA HO	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Merit
LUI KA LAI	M SPRING SCHOOL	Merit
CHU TSZ POK	M VENUS COLLEGE	Merit
CARLOS MORGAN	M WINTER SECONDARY SCHOOL	Merit
ANTHONY PETERSON	M MARS COLLEGE	Merit
FRANK JONES	M SUN MEMORIAL SCHOOL	Merit
SKYLER POON	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Merit
DOUGLAS BAKER	M FALL SECONDARY SCHOOL	Merit

## Multiple-Choice Analysis Report

MARTIN PHILLIPS	M MARS COLLEGE	Merit
BRIAN JAMES	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Merit
EDWARD MURPHY	M LEAGUE SCHOOL	Merit
ALBERT DIAZ	M FALL SECONDARY SCHOOL	Merit
LEE WING HAN	F WINTER SECONDARY SCHOOL	Credit
KELLY GONZALEZ	F MARS COLLEGE	Credit
BETTY GARCIA	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Credit
RACHEL PERRY	F EARTH COLLEGE	Credit
LAURA ADAMS	F SPRING SCHOOL	Credit
IRENE BROWN	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Credit
DIANA WARD	F MOON SECONDARY SCHOOL	Credit
GLORIA PRICE	F WINTER SECONDARY SCHOOL	Credit
DONNA WALKER	F SUN MEMORIAL SCHOOL	Credit
MICHELLE CLARK	F SUN MEMORIAL SCHOOL	Credit
KAREN WILLIAMS	F SUN MEMORIAL SCHOOL	Credit
JULIA YOUNG	F MARS COLLEGE	Credit
KATHERINE MORGAN	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Credit
CHAN SIN LING CHARLOTTE	F MARS COLLEGE	Credit
LAU MING LEE CLARRISA	F WINTER SECONDARY SCHOOL	Credit
FUNG MO YEE	F SPRING SCHOOL	Credit
TAM WING SUM	F SPRING SCHOOL	Credit
NG CHI YANF	F LEAGUE SCHOOL	Credit
TSANG MAN KEI	F SPRING SCHOOL	Credit
LIU CHUN YEE	M SUN MEMORIAL SCHOOL	Credit
TSE KIN LOK	M VENUS COLLEGE	Credit
FUNG KWOK LUN	M MAPLE TREE COLLEGE	Credit
WONG KAM SHING	M WINTER SECONDARY SCHOOL	Credit
YIP WAI TING	M MARS COLLEGE	Credit
VICTOR COLLINS	M FALL SECONDARY SCHOOL	Credit
WALTER ROGERS	M MARS COLLEGE	Credit
BRANDON MARTINEZ	M WINTER SECONDARY SCHOOL	Credit
AARON COOK	M SUN MEMORIAL SCHOOL	Credit
ANNIE RODRIGUEZ	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Credit
CAROLYN GREEN	F SUN MEMORIAL SCHOOL	Credit

Key: HD=High Distinction, D=Distinction

School award winners of the competition



## Multiple-Choice Analysis Report

\*\*\*\*\*

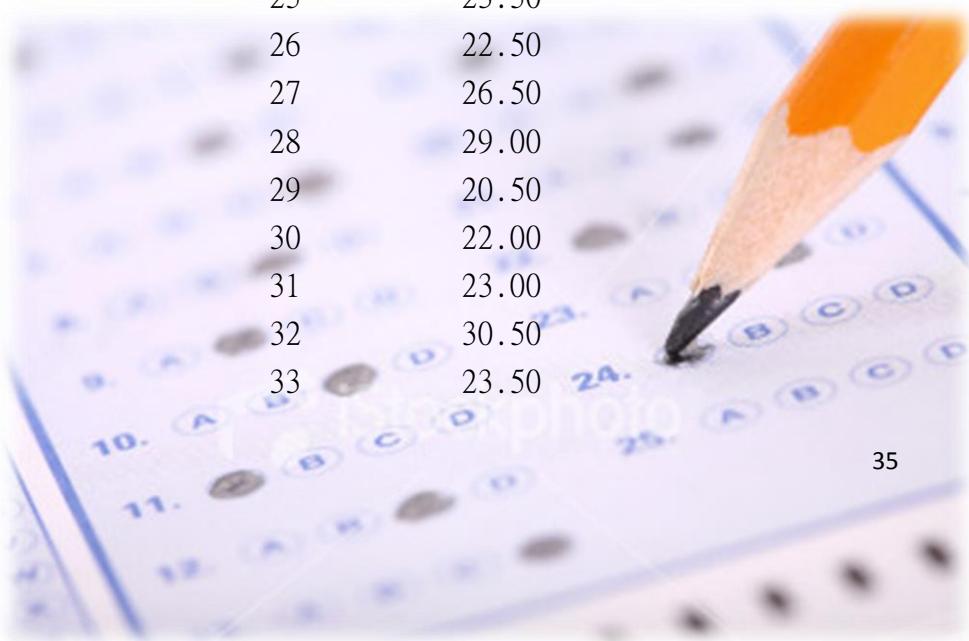
School	Total sum of score	Award
*****		
**		
HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	298	Champion
WINTER SECONDARY SCHOOL	233	1st Runner Up
SUN MEMORIAL SCHOOL	224	2nd Runner Up
VENUS COLLEGE	201	Participation
FALL SECONDARY SCHOOL	181	Participation
MARS COLLEGE	174	Participation
SPRING SCHOOL	173	Participation
MAPLE TREE COLLEGE	156	Participation
EARTH COLLEGE	153	Participation
MOON SECONDARY SCHOOL	145	Participation
LEAGUE SCHOOL	125	Participation
ATMOSPHERE COLLEGE	12	Participation
FIRE SCHOOL	7	Participation
SUMMER SCHOOL	3	Participation

## QUESTIONS AND SCORE ANALYSIS:

Percentage correct for each questions

\*\*\*\*\*

Question No.	% correct	Question No.	% correct
1	31.50	21	25.50
2	27.00	22	24.00
3	22.50	23	29.50
4	23.50	24	34.00
5	23.50	25	23.50
6	24.50	26	22.50
7	23.50	27	26.50
8	32.00	28	29.00
9	25.00	29	20.50
10	25.50	30	22.00
11	25.50	31	23.00
12	25.50	32	30.50
13	27.00	33	23.50



## Multiple-Choice Analysis Report

14	23.50	34	21.50
15	26.50	35	28.50
16	28.00	36	29.00
17	24.50	37	26.00
18	25.50	38	25.00
19	29.50	39	26.00
20	29.00	40	29.50

Standard deviation of score is: 29.57

Mean: 10.43

Median: 10

Range: 37

Upper quartile :12

Lower quartile :8

Inter-quartile range :4

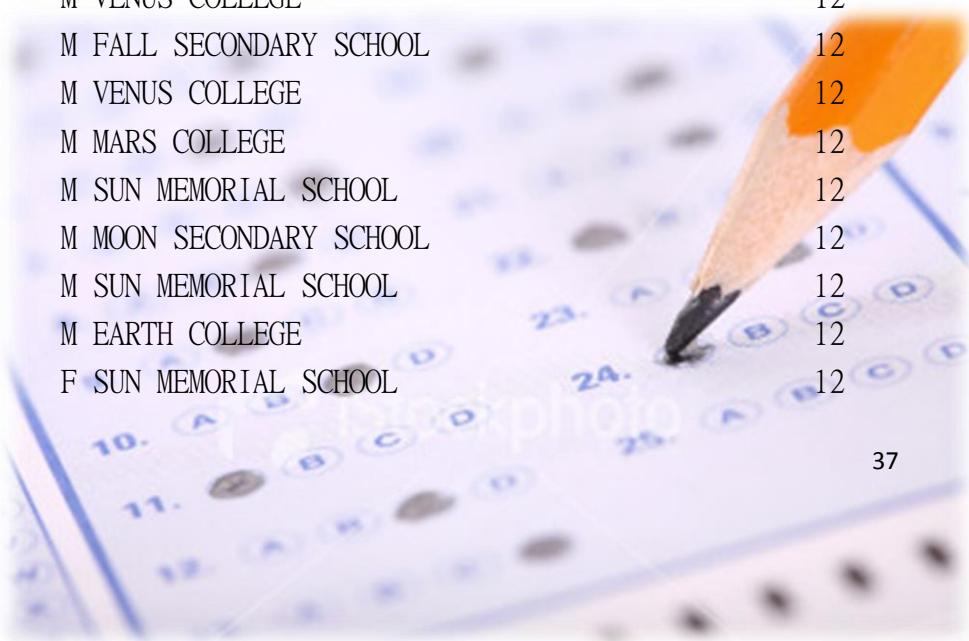
The list of all participants

\*\*\*\*\*

Name	Sex	School	Score
LI KA SHING	M	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	40
NG SIU YEE	F	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	35
LEUNG HIU FUNG	M	SPRING SCHOOL	32
HOWARD YOUNG	M	FALL SECONDARY SCHOOL	18
SO PANG HIN	M	MOON SECONDARY SCHOOL	17
ADAM HENDERSON	M	VENUS COLLEGE	17
CHAN TIN WAI	M	WINTER SECONDARY SCHOOL	16
FONG CHEUK HO	M	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	16
CHAN PUI HAN	F	MOON SECONDARY SCHOOL	16
CYNTHIA MARTIN	F	VENUS COLLEGE	16
BRENDA RIVERA	F	MARS COLLEGE	16
JANICE BROOKS	F	VENUS COLLEGE	16
TAM WING YAN	F	FALL SECONDARY SCHOOL	16

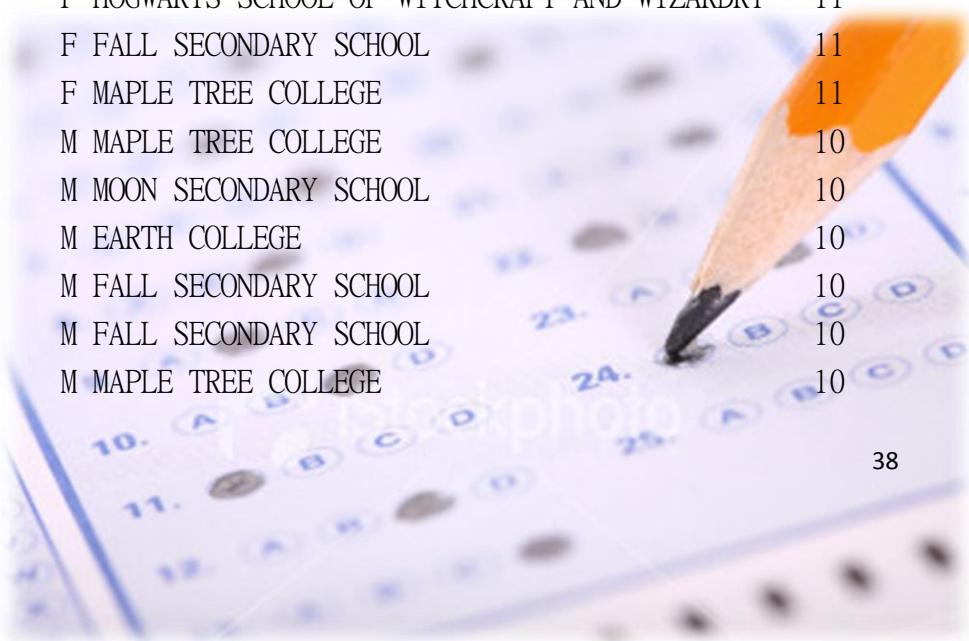
## Multiple-Choice Analysis Report

CARL ALEXANDER	M MAPLE TREE COLLEGE	15
TERESA JENKINS	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	15
WONG NGA YIN BELINDA	F EARTH COLLEGE	15
LEE SHU JEN	M LEAGUE SCHOOL	14
LEUNG WAI CHI	M EARTH COLLEGE	14
LUI KEI KWAN	M LEAGUE SCHOOL	14
RAYMOND COX	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	14
SENA KOBAYAKAWA	M MAPLE TREE COLLEGE	14
MARYLYN RUSSELL	F VENUS COLLEGE	14
KATHY BAKER	F SUN MEMORIAL SCHOOL	14
CHAN WING YEE	F SPRING SCHOOL	14
AU WING HING	M FALL SECONDARY SCHOOL	13
CHU KIN FUNG	M WINTER SECONDARY SCHOOL	13
FONG KA WING	M FALL SECONDARY SCHOOL	13
LAM MAN CHUNG	M MAPLE TREE COLLEGE	13
LAW HO YU	M FALL SECONDARY SCHOOL	13
MAR TIN PONG	M SUN MEMORIAL SCHOOL	13
NG WAI CHUNG	M WINTER SECONDARY SCHOOL	13
GARY WOOD	M MAPLE TREE COLLEGE	13
TAYLOR WASHINGTON	M WINTER SECONDARY SCHOOL	13
GREGORY LONG	M MARS COLLEGE	13
LEUNG HOI YAN	F VENUS COLLEGE	13
DEBRA COX	F FALL SECONDARY SCHOOL	13
RUBY LOPEZ	F MOON SECONDARY SCHOOL	13
VIRGINIA PATTERSON	F SUN MEMORIAL SCHOOL	13
ANN WATSON	F WINTER SECONDARY SCHOOL	13
TONG YIU YAN	F EARTH COLLEGE	13
YU JIN REN	M ATMOSPHERE COLLEGE	12
HO HOI TO	M WINTER SECONDARY SCHOOL	12
CHAN HEUNG WAI	M VENUS COLLEGE	12
AU CHEUK KEI	M FALL SECONDARY SCHOOL	12
TSANG MAU YIU	M VENUS COLLEGE	12
ERIC COOPER	M MARS COLLEGE	12
PHILIP SMITH	M SUN MEMORIAL SCHOOL	12
JOSHUA THOMPSON	M MOON SECONDARY SCHOOL	12
BRUCE CARTER	M SUN MEMORIAL SCHOOL	12
DAIKICHI KOMUSUBI	M EARTH COLLEGE	12
CHEUNG HOI YAN	F SUN MEMORIAL SCHOOL	12



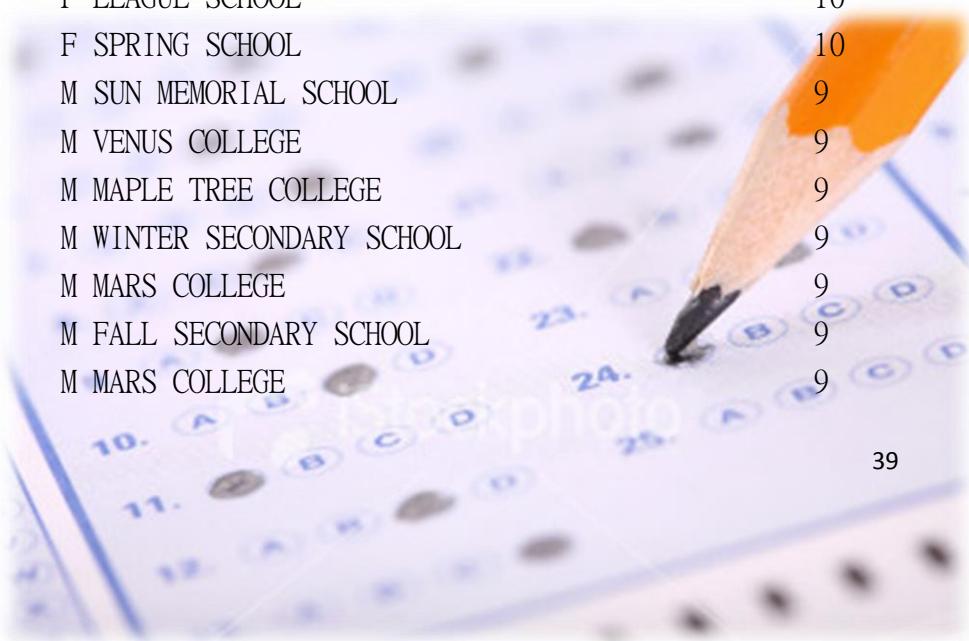
## Multiple-Choice Analysis Report

NORMA BELL	F VENUS COLLEGE	12
LISA WRIGHT	F EARTH COLLEGE	12
MARGARET BENNETT	F WINTER SECONDARY SCHOOL	12
KATHLEEN PEREZ	F WINTER SECONDARY SCHOOL	12
CHING TSAN WING	F MAPLE TREE COLLEGE	12
CHAN KA YI	F WINTER SECONDARY SCHOOL	12
YEUNG CHING YIU	F SPRING SCHOOL	12
CHAN TAI MAN	M SPRING SCHOOL	11
WONG KA LOK	M LEAGUE SCHOOL	11
CHAN SAU TIN JUSTIN	M EARTH COLLEGE	11
CHEUNG CHUN TING	M SUN MEMORIAL SCHOOL	11
CHEUNG WAI KEI	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	11
LAI KA WING	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	11
TO KENNETH	M SPRING SCHOOL	11
YIP KIM LUNG	M WINTER SECONDARY SCHOOL	11
CHAN PAK YIN	M MARS COLLEGE	11
LAM KIN CHUN	M EARTH COLLEGE	11
WONG SZE HO STANLEY	M SPRING SCHOOL	11
EUGENE LEE	M MOON SECONDARY SCHOOL	11
BOBBY PARKER	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	11
JESSE BAILEY	M WINTER SECONDARY SCHOOL	11
AMPERE HUGHES	M MOON SECONDARY SCHOOL	11
JOSE CLARK	M VENUS COLLEGE	11
DAVID FLORES	M VENUS COLLEGE	11
JOE KING	M MOON SECONDARY SCHOOL	11
WONG SIU YAN	F SPRING SCHOOL	11
NG YEE KIU	F MAPLE TREE COLLEGE	11
MARTHA MARTINEZ	F WINTER SECONDARY SCHOOL	11
NANCY DIAZ	F MAPLE TREE COLLEGE	11
DEBORAH GRAY	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	11
NICOLE JACKSON	F FALL SECONDARY SCHOOL	11
PHYLLIS LEWIS	F MAPLE TREE COLLEGE	11
CHAN KA KIN	M MAPLE TREE COLLEGE	10
MICHAEL JACKSON	M MOON SECONDARY SCHOOL	10
LAI KA WING	M EARTH COLLEGE	10
MOK SIU HEI	M FALL SECONDARY SCHOOL	10
WONG KA LOK	M FALL SECONDARY SCHOOL	10
CHU WAI HON	M MAPLE TREE COLLEGE	10



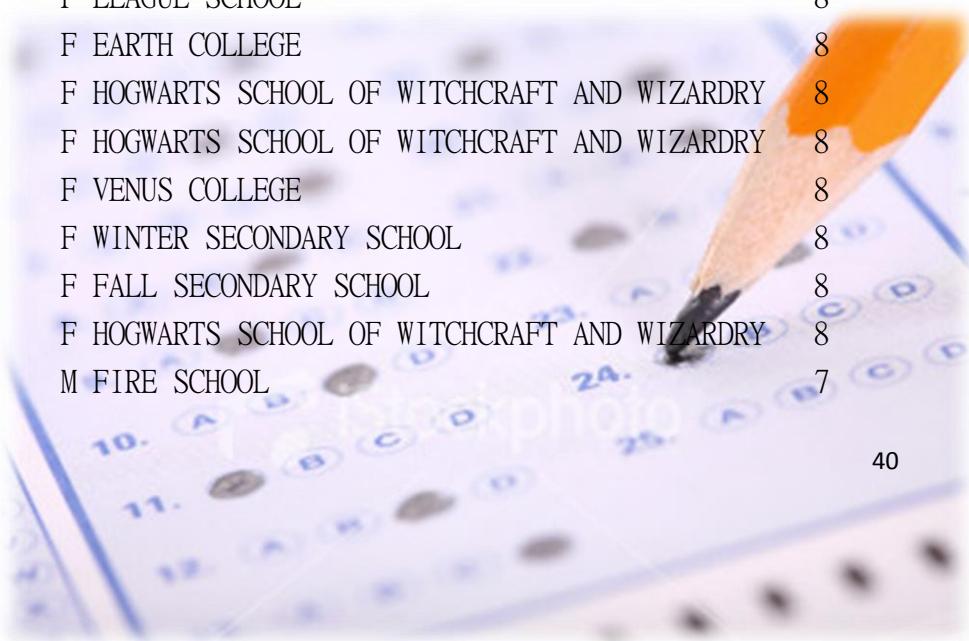
## Multiple-Choice Analysis Report

LUI KA HO	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	10
LUI KA LAI	M SPRING SCHOOL	10
CHU TSZ POK	M VENUS COLLEGE	10
CARLOS MORGAN	M WINTER SECONDARY SCHOOL	10
ANTHONY PETERSON	M MARS COLLEGE	10
FRANK JONES	M SUN MEMORIAL SCHOOL	10
SKYLER POON	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	10
DOUGLAS BAKER	M FALL SECONDARY SCHOOL	10
MARTIN PHILLIPS	M MARS COLLEGE	10
BRIAN JAMES	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	10
EDWARD MURPHY	M LEAGUE SCHOOL	10
ALBERT DIAZ	M FALL SECONDARY SCHOOL	10
LEE WING HAN	F WINTER SECONDARY SCHOOL	10
KELLY GONZALEZ	F MARS COLLEGE	10
BETTY GARCIA	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	10
RACHEL PERRY	F EARTH COLLEGE	10
LAURA ADAMS	F SPRING SCHOOL	10
IRENE BROWN	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	10
DIANA WARD	F MOON SECONDARY SCHOOL	10
GLORIA PRICE	F WINTER SECONDARY SCHOOL	10
DONNA WALKER	F SUN MEMORIAL SCHOOL	10
MICHELLE CLARK	F SUN MEMORIAL SCHOOL	10
KAREN WILLIAMS	F SUN MEMORIAL SCHOOL	10
JULIA YOUNG	F MARS COLLEGE	10
KATHERINE MORGAN	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	10
CHAN SIN LING CHARLOTTE	F MARS COLLEGE	10
LAU MING LEE CLARRISA	F WINTER SECONDARY SCHOOL	10
FUNG MO YEE	F SPRING SCHOOL	10
TAM WING SUM	F SPRING SCHOOL	10
NG CHI YANF	F LEAGUE SCHOOL	10
TSANG MAN KEI	F SPRING SCHOOL	10
LIU CHUN YEE	M SUN MEMORIAL SCHOOL	9
TSE KIN LOK	M VENUS COLLEGE	9
FUNG KWOK LUN	M MAPLE TREE COLLEGE	9
WONG KAM SHING	M WINTER SECONDARY SCHOOL	9
YIP WAI TING	M MARS COLLEGE	9
VICTOR COLLINS	M FALL SECONDARY SCHOOL	9
WALTER ROGERS	M MARS COLLEGE	9



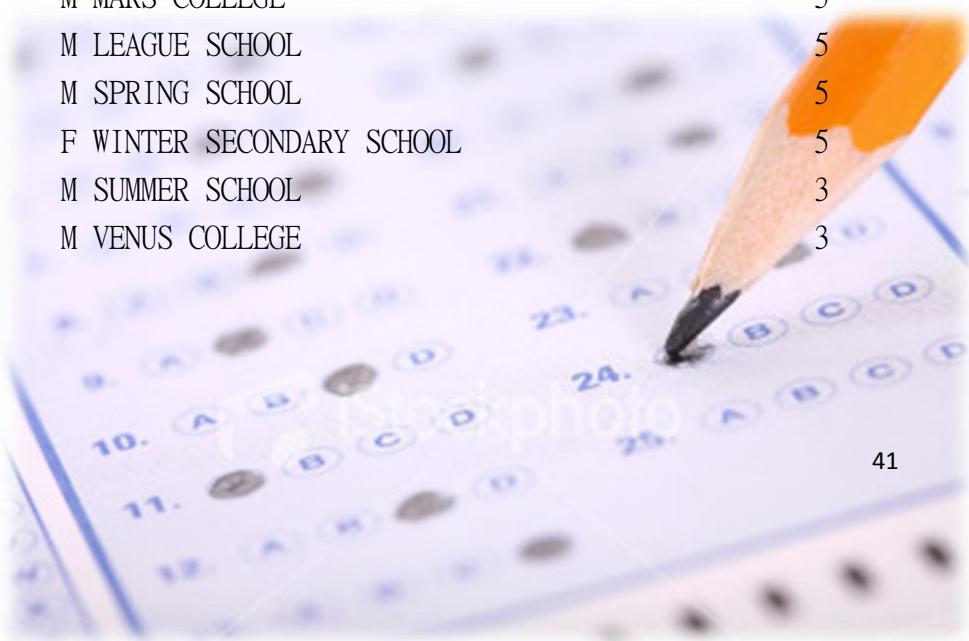
## Multiple-Choice Analysis Report

BRANDON MARTINEZ	M WINTER SECONDARY SCHOOL	9
AARON COOK	M SUN MEMORIAL SCHOOL	9
ANNIE RODRIGUEZ	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	9
CAROLYN GREEN	F SUN MEMORIAL SCHOOL	9
JULIE STEWART	F SUN MEMORIAL SCHOOL	9
CHAN MAN HA	F MARS COLLEGE	9
LEE LOK YIU	F MARS COLLEGE	9
LAU WAI HING	F LEAGUE SCHOOL	9
SUEN KA YIUF	F SUN MEMORIAL SCHOOL	9
WONG TUNG TUNG	M VENUS COLLEGE	8
CHAN CHUN HUNG	M EARTH COLLEGE	8
LAW CHI HO	M MARS COLLEGE	8
YAU WAI HIN	M FALL SECONDARY SCHOOL	8
CHU PUI HO	M MAPLE TREE COLLEGE	8
PANG LONG WING	M SUN MEMORIAL SCHOOL	8
RUSSELL ADAMS	M EARTH COLLEGE	8
JOHN HARRIS	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	8
GAREN JOHNSON	M VENUS COLLEGE	8
ALBERT EINSTEIN	M SUN MEMORIAL SCHOOL	8
TODD MARTIN	M MARS COLLEGE	8
PAUL BROOKS	M MOON SECONDARY SCHOOL	8
SEAN RIVERA	M SPRING SCHOOL	8
YOICHI HIRUMA	M SUN MEMORIAL SCHOOL	8
NATSUHIKO TAKI	M MARS COLLEGE	8
CHAN KA YI	F LEAGUE SCHOOL	8
SUSAN ROBINSON	F EARTH COLLEGE	8
JUDY TORRES	F SUN MEMORIAL SCHOOL	8
SHIRLEY CARTER	F SPRING SCHOOL	8
ANNE FOSTER	F VENUS COLLEGE	8
MARIA LEE	F LEAGUE SCHOOL	8
LINDA WASHINGTON	F EARTH COLLEGE	8
BEVERLY HOWARD	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	8
LOIS PARKER	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	8
SHANICE POON	F VENUS COLLEGE	8
TSANG HIU MAN	F WINTER SECONDARY SCHOOL	8
MAU YAN WING	F FALL SECONDARY SCHOOL	8
WONG SHUN YAN	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	8
WONG TIN LONG	M FIRE SCHOOL	7



## Multiple-Choice Analysis Report

SEID YUK FAI SUNNYSON	M FALL SECONDARY SCHOOL	7
LEUNG CHUN HO	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	7
WONG CHUNG KIT	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	7
CHAN HON YIN	M EARTH COLLEGE	7
SO KA YIN	M SUN MEMORIAL SCHOOL	7
JUSTIN PEREZ	M WINTER SECONDARY SCHOOL	7
JONATHAN LOPEZ	M SUN MEMORIAL SCHOOL	7
JIMMY PERRY	M LEAGUE SCHOOL	7
ERNEST WILLIAMS	M MARS COLLEGE	7
RYOKAN KURITA	M MOON SECONDARY SCHOOL	7
CHAN PUI KI	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	7
HEATHER JAMES	F MAPLE TREE COLLEGE	7
CATHERINE COLLINS	F MOON SECONDARY SCHOOL	7
IP KA WING	F LEAGUE SCHOOL	7
LIU YUEN WAI	F VENUS COLLEGE	7
LAM CHI HO	M VENUS COLLEGE	6
CHAN TIN WAI	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	6
CHENG YU LAP	M SUN MEMORIAL SCHOOL	6
JACK TAYLOR	M MOON SECONDARY SCHOOL	6
JEREMY GARCIA	M MAPLE TREE COLLEGE	6
RONALD REED	M LEAGUE SCHOOL	6
RALPH DAVIS	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	6
PAMELA WOOD	F MOON SECONDARY SCHOOL	6
JANE JOHNSON	F LEAGUE SCHOOL	6
PATRICIA KING	F WINTER SECONDARY SCHOOL	6
BONNIE PHILLIPS	F MAPLE TREE COLLEGE	6
ANGELA COLEMAN	F EARTH COLLEGE	6
LEUNG CHEUK WING	M LEAGUE SCHOOL	5
LI SI XIAN	M LEAGUE SCHOOL	5
SO TING PONG	M MARS COLLEGE	5
WILLIAM SANCHEZ	M LEAGUE SCHOOL	5
HAROLD EVANS	M SPRING SCHOOL	5
ANDREA EVANS	F WINTER SECONDARY SCHOOL	5
WONG SAI MAN	M SUMMER SCHOOL	3
CHU TSZ HIM	M VENUS COLLEGE	3



# **Chapter 4 Implementation**

## **4.1 Brief Description**

In this chapter, the followings of the implementation of the MC Analysis program will be discussed:

- ⌚ The **functions** that will be performed by each procedure in the program.
- ⌚ The **data structures** that will be used in the program.
- ⌚ The main **algorithms** used in the program
- ⌚ Some of the **program codes**
- ⌚ The **user interface** and **analysis report**.

## **4.2 Procedures in the Program**

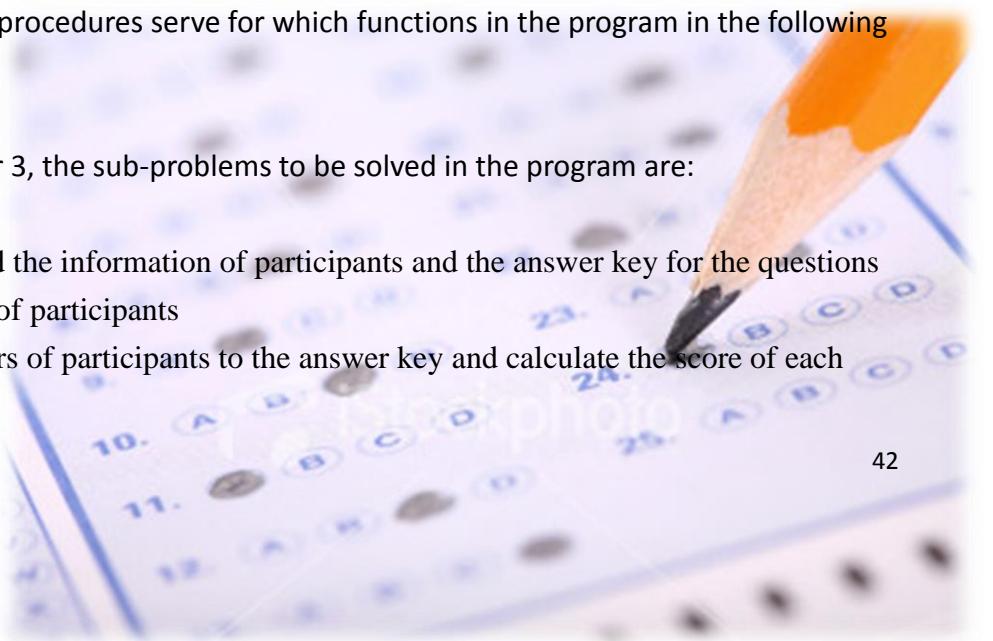
As stated in [Chapter 3](#), several sub-problems need to be solved in the program, for this the following procedures will be constructed in the program:

1. welcome;
2. readinput;
3. readschool;
4. readdisplay;
5. calscore;
6. sorts;
7. sortp;
8. mainmenu;

I will further explain which procedures serve for which functions in the program in the following few pages.

Again, according to Chapter 3, the sub-problems to be solved in the program are:

- ⌚ To greet the user
- ⌚ To read the answer and the information of participants and the answer key for the questions and count the number of participants
- ⌚ To compare the answers of participants to the answer key and calculate the score of each participant



## Multiple-Choice Analysis Report

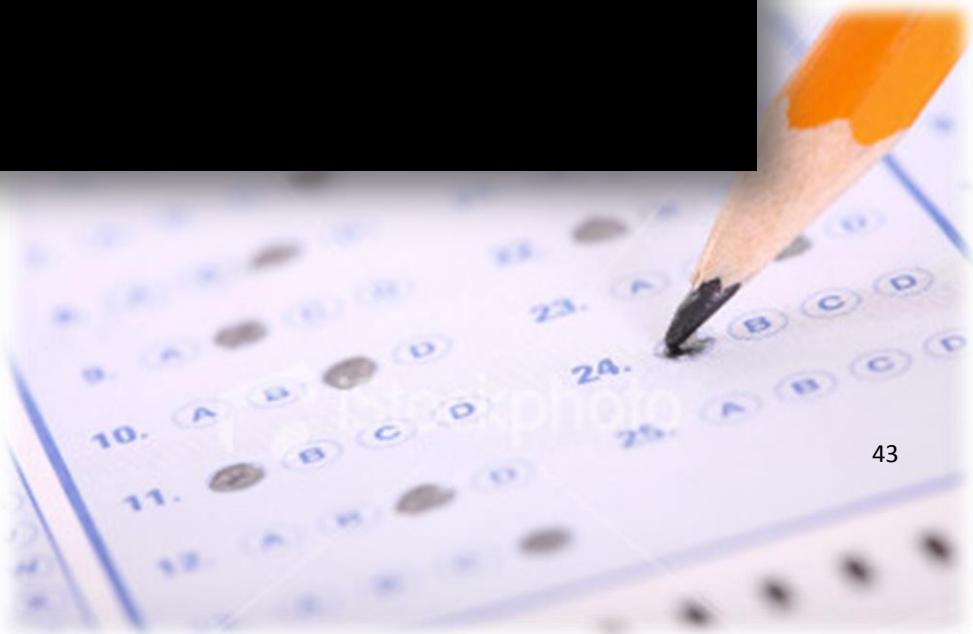
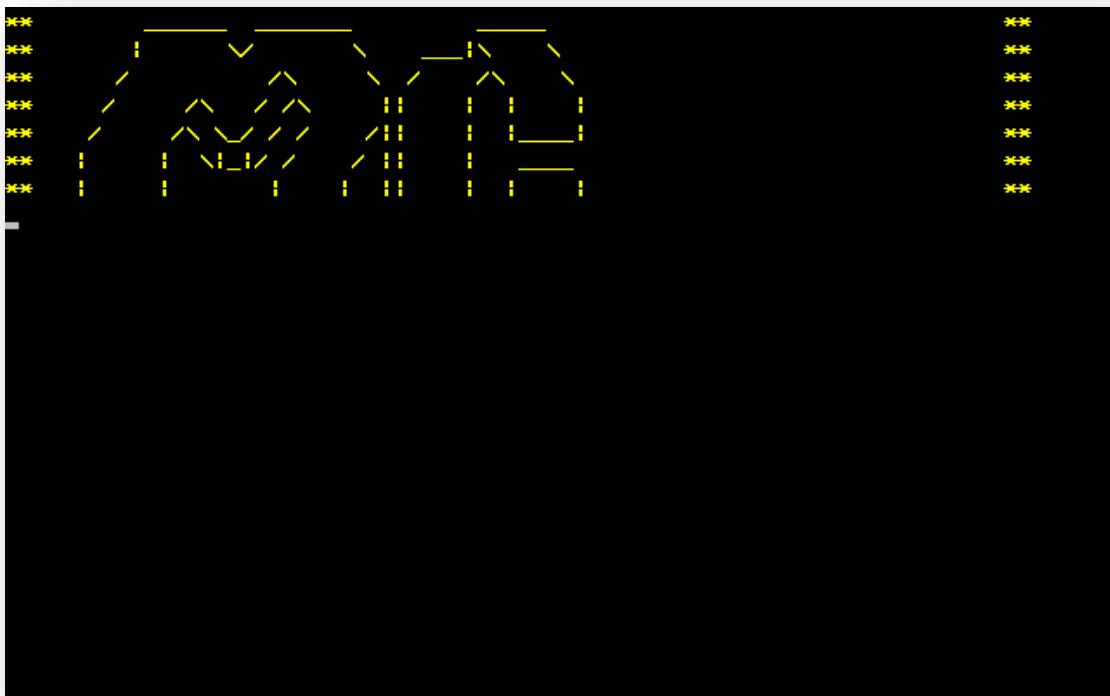
- (C) To sort the records
- (C) To calculate the results of each school
- (C) To find out the award winners
- (C) To analyze questions and calculate for the statistics
- (C) To display results on the screen
- (C) To search for a particular participant
- (C) To save the report in text form for users

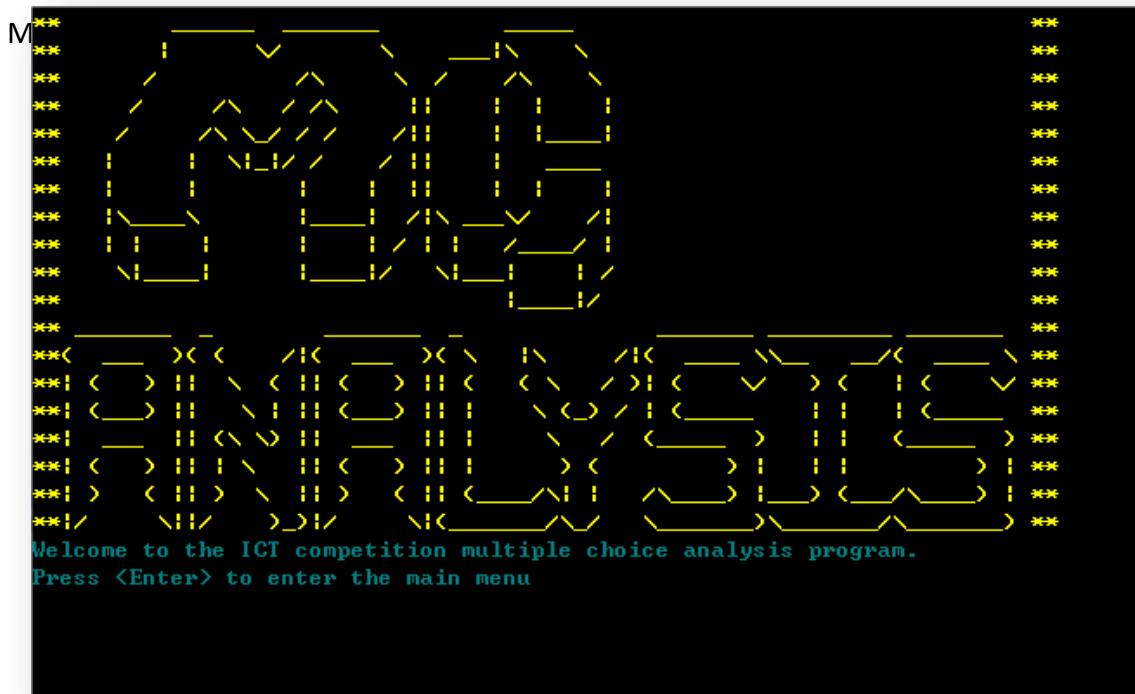
So, the procedures are designed to be able to solve these problems.

### 1. To greet the user

A **welcoming screen** is included in the procedure `welcome`. This is to greet the user in a friendly way such that the user feels **comfortable** to use the program. I duplicated an ASCII art and use colours to decorate the welcoming screen to make it more **decent** and **beautiful**.

Also, the **delay** function from the **crt library** is also used to modify an **animated effect of “ringing down the curtain”**, which makes the screen more **attractive**.





## Screen shots of the welcoming screen

Program codes of the welcoming screen:

```
{1}procedure welcome;
begin;
textcolor(14);
writeln('** _____ _____ _____ *');
delay(100);
writeln('** | \ / \ \_ | \ \ / *');
delay(100);
writeln('** / \ / \ \ / \ / \ / \ *');
delay(100);
writeln('** / \ \_ / / / / | | | | | *');
delay(100);
writeln('** | | \_ | / / / | | | | | *');
delay(100);
writeln('** | | | | | | | | | | | *');
delay(100);
writeln('** | \____\ | ____| / \____\ / / | *');
delay(100);
writeln('** | | | | | | | | | | | | *');
```

## Multiple-Choice Analysis Report



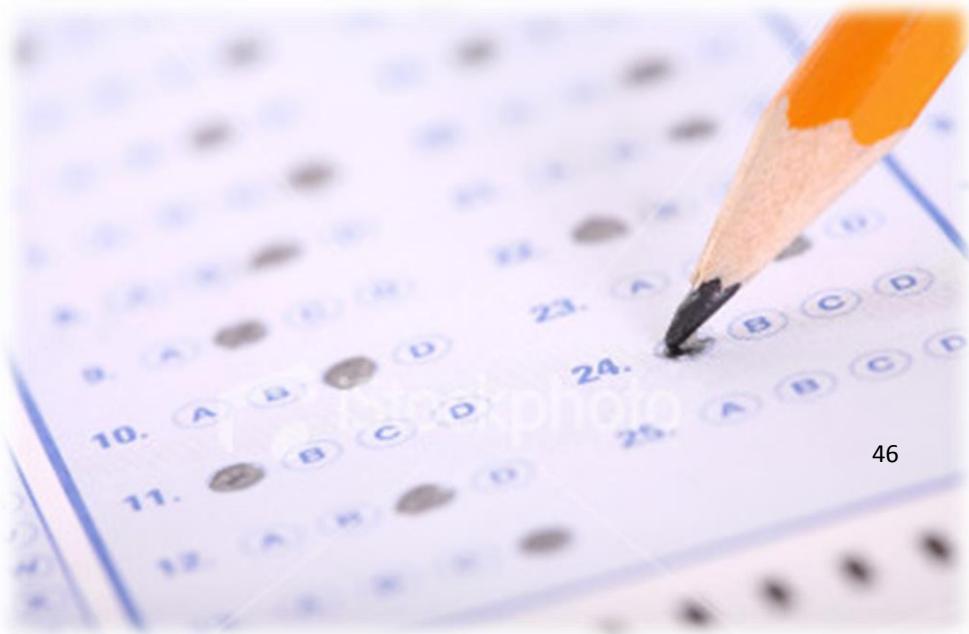
## 2. To read the answer and the information of participants

### and the answer key for the questions

A procedure called `readininput` is written to **read** the **answer**, and the **information of participants** and the **answer key** for the questions

The answer and the information of participants are stored in **one single file**, which its file name is inputted by the user as mentioned. Then the program will read the **each specific field** of participants one by one by the **'copy' function** of strings into the **array** until the file ends. Similarly, the school names and the answer key will be read into the **array** and the **string** defined for storing the school names and answers for questions. A **while loop** with condition to terminate the loop as **eof** is used instead of using program counters because we can ensure **all the data** can be written from the file. A program counter, **n** is included for counting the **total number of participants** in the procedure .

While reading the participants' sex, counters are also used to **count for the number of male participants (m)** and **female (f) participants**, because it can be done easily while reading in the data.

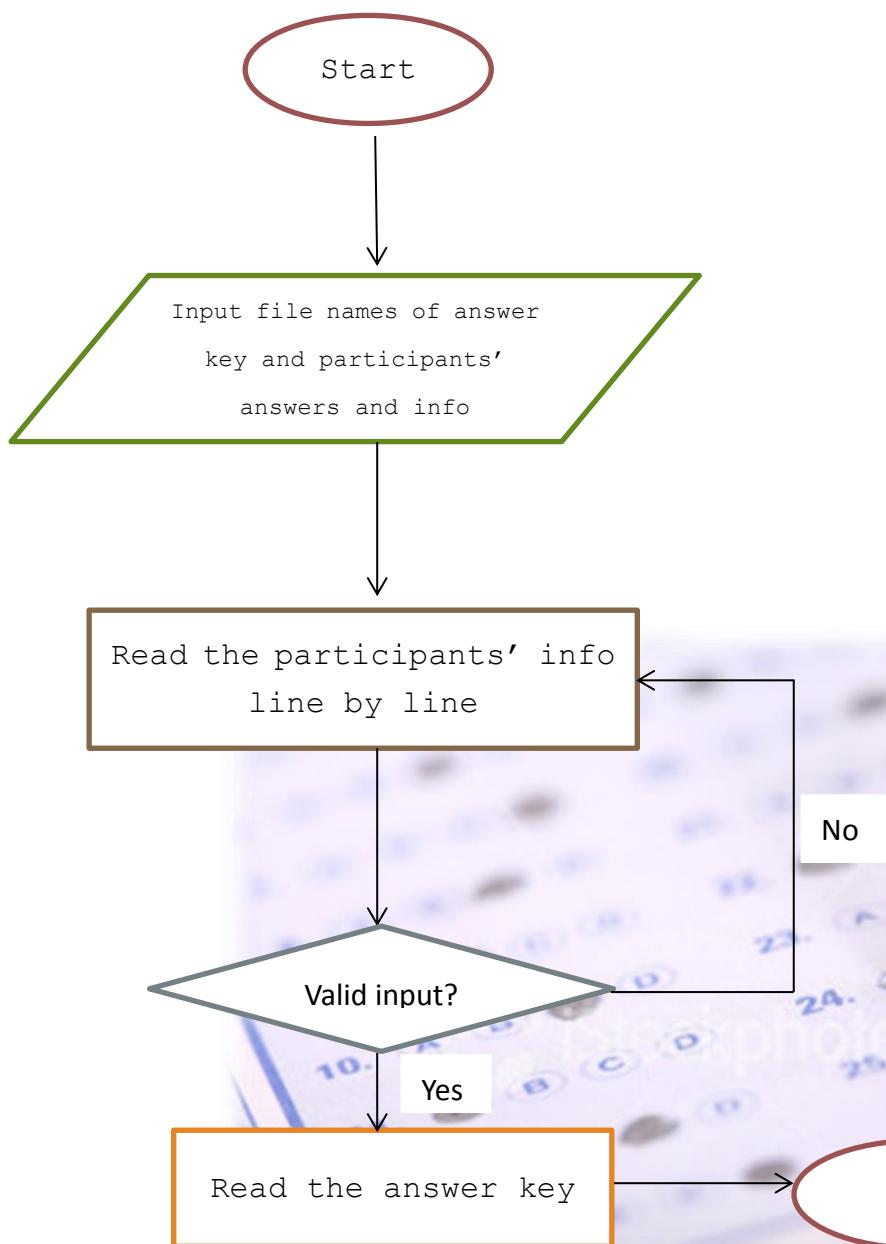


## Multiple-Choice Analysis Report

Again, ASCII art is added to the page of inserting data for decoration purpose.

```
8 8888 ,o888888o.8888888 8888888888  
8 8888 8888 `88. 8 8888  
8 8888 ,8 8888 `8. 8 8888  
8 8888 88 8888 8 8888  
8 8888 88 8888 8 8888  
8 8888 88 8888 8 8888  
8 8888 88 8888 8 8888  
8 8888 `8 8888 .8' 8 8888  
8 8888 8888 ,88 8 8888  
8 8888 `8888888P 8 8888
```

A flowchart below can illustrate the procedure in a simple manner.



## Multiple-Choice Analysis Report

Before reading the file, **validation** to the file path is needed. By the principle of **garbage-in-garbage-out(GIGO)**, invalid file path will cause **invalid information** or **self-termination** of the program. For this, **compiler derivatives** are introduced to check whether **the file exists**.

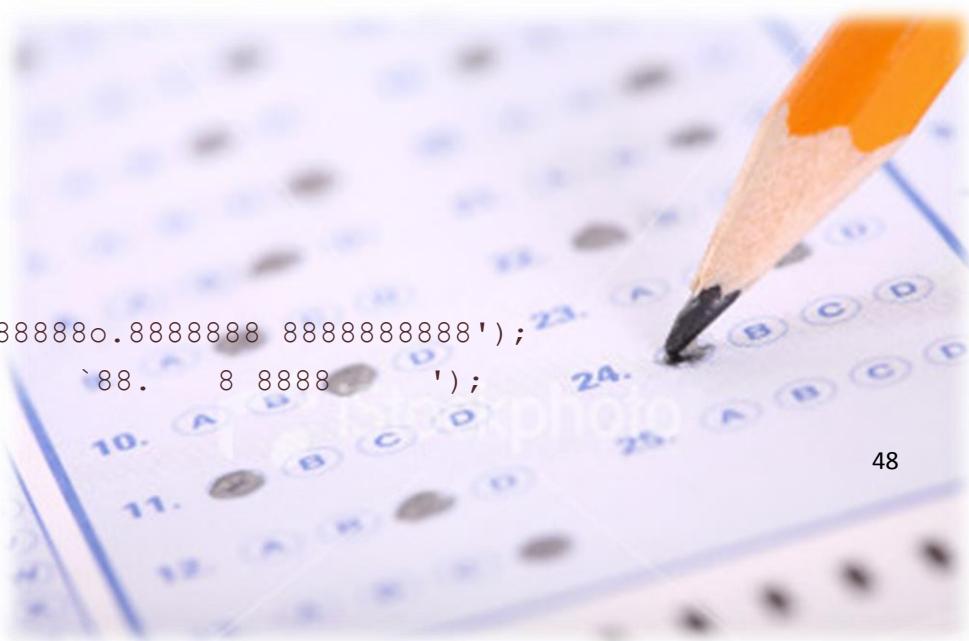
System function `IOResult` will return 0 if the file can be successfully opened, else, a statement that telling users to correct their input will be shown, the same is done to the reading of answer key.

```
assign(infile,pinfo); //checking whether the file inputted exists
{$I-} { disable i/o error checking }
reset(infile);
{$I+} { enable again i/o error checking}
if (IOResult <> 0) then
begin
    textcolor(3);
    writeln('The file required to be opened is not
found.');
    readln;
end
else begin
    flag:=true ;
    close(infile);
end

until flag=true;
```

Program codes of the procedure:

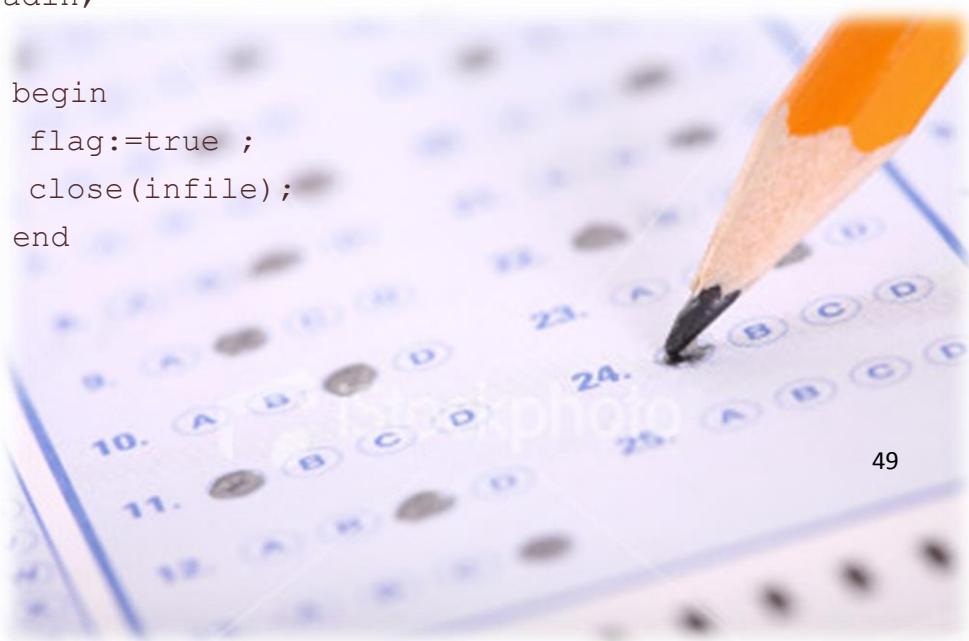
```
{2}procedure readinput;
const max=200;
var infile: text;
pinfo,line: string;
flag:boolean;
begin
clrscr;
writeln(' 8 8888     ,08888880.8888888 888888888');
writeln(' 8 8888 8888 `88.   8 8888      '');
```



## Multiple-Choice Analysis Report

```
writeln(' 8 8888 ,8 8888     `8.    8 8888      ');
writeln(' 8 8888 88 8888          8 8888      ');
writeln(' 8 8888 `8 8888       .8    8 8888      ');
writeln(' 8 8888     8888     ,88    8 8888      ');
writeln(' 8 8888     `88888888P    8 8888      COMPETITION MC ANALYSIS');

writeln('*****');
repeat
  repeat
    textColor(3);
    writeln('Please input the file name of the participants'' information
and answers');
    write('default:(MCDATA.txt) :');
    textColor(15);           {reading the participants' info}
    flag:=false;
    readln(pinfo);
    assign(infile,pinfo);   //checking whether the file inputted exists
    {$I-}   { disable i/o error checking }
    reset(infile);
    {$I+}   { enable again i/o error checking}
    if (IOResult <> 0) then
      begin
        textColor(3);
        writeln('The file required to be opened is not
found.');
        readln;
      end
    else begin
      flag:=true ;
      close(infile);
    end
  until flag=true;
```



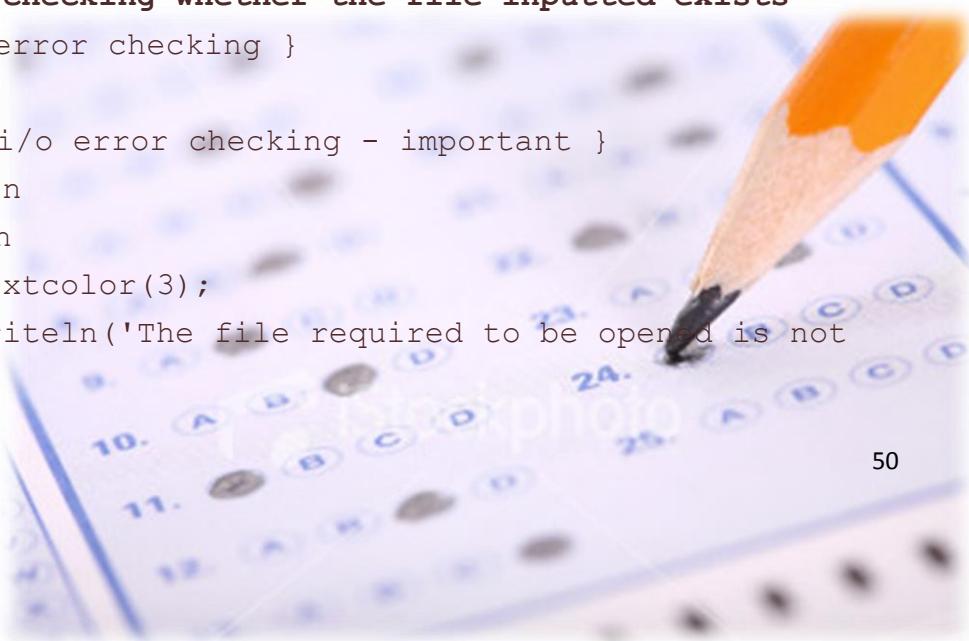
## Multiple-Choice Analysis Report

```
assign(infile,pinfo);
reset(infile);
n:=0;

while not eof(infile) do begin
  n:=n+1;
  Readln(infile,line);
  pname[n]:=copy(line,1,25);
  sex[n]:=line[31];
  pschool[n]:=copy(line,33,45);
  pans[n]:=copy(line,83,40);
  if sex[n]='M' then m:=m+1; {counting male participants}
  if sex[n]='F' then f:=f+1; {counting female participants}

end;
close(infile); {end of reading the participants' info}
```

```
repeat
  repeat
    textColor(3);
    writeln('Please input the file name of the answer key');
    write('default:(ANSKEY.txt) :');
    textColor(15); {reading the anskey}
    flag:=false;
    readln(ans);
    assign(infile,ans); //checking whether the file inputted exists
    {$I-} { disable i/o error checking }
    reset(infile);
    {$I+} { enable again i/o error checking - important }
    if (IOResult <> 0) then
      begin
        textColor(3);
        writeln('The file required to be opened is not
found.');
```



## Multiple-Choice Analysis Report

```
        readln;
    end
else
begin
    flag:=true ;
    close(infile);
end;
until flag=true;

textcolor(15);
assign(infile,ans);      {reading the anskey}
reset(infile);
readln(infile,ans);
close(infile);           {end of reading the anskey}
```

Another procedure, `readschool` is used to read all the school names which the participants are in. Similar to the previous procedure, the names are read line by line.

Also a counter is included to count for the number of participants in each school (`snos[i]`). An array parallel to the one storing school names is used (`school[i]`), therefore **the number of participants in each school and their school names would be consistent**.

Similarly, the **same method** is adopted in this procedure to **validate** the input of school names.

Program codes of the procedure:

```
{3}procedure readschool;
var  infile: text;
line: string;
i,j:integer;
flag:boolean;
begin
repeat
repeat
textcolor(3);
writeln('Please input the file name of school list');
write('default:(SCHOOL.txt) :');
```



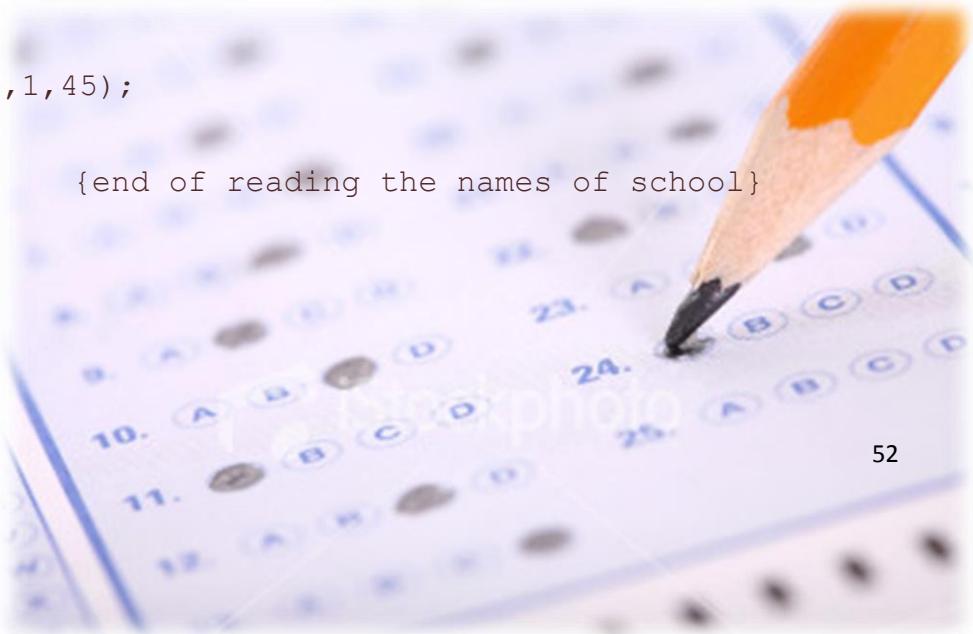
## Multiple-Choice Analysis Report

```
textcolor(15);           {reading the participants' info}
flag:=false;
readln(schoollist);
assign(infile,schoollist); //checking whether the file inputted exists
{$I-}   { disable i/o error checking }
reset(infile);
{$I+}   { enable again i/o error checking}
if (IOResult <> 0) then
begin
    textcolor(3);
    writeln('The file required to be opened is not
found.');
    readln;
end
else
begin
    flag:=true ;
    close(infile);
end;
until flag=true;

textcolor(3);           {reading the names of school}

textcolor(15);
assign(infile,schoollist);
reset(infile);
nos:=0;
while not eof(infile) do begin
    nos:=nos+1;           {counting the number of participants in each
school}
    readln(infile,line);
    school[nos]:=copy(line,1,45);
end;
close(infile);          {end of reading the names of school}

for i:= 1 to n do
begin
    for j:=1 to nos do
```



## Multiple-Choice Analysis Report

```
if pschool[i]=school[j] then snos[j]:=snos[j]+1;  
end;  
end;
```

### 3. To compare the answers of participants to the answer key and calculate score

A procedure called `calscore` is written to **compare the answers of participants to the answer key** so as to calculate the score of each participant, total number of participants who answer correctly for each question, the score of each school and the mean score.

Firstly, an **IF statement** is used to count the **score of each participant** and the **total number of participants**.

```
if (pans[j][k]=ans[k]) then begin  
    pscore[j]:=pscore[j]+1;  
    qscore[k]:=qscore[k]+1;  
end;
```

In simple words, if the answer of the participant for one particular question is equal to the key of that question, the score of the participant (`pscore[j]`) will be incremented by 1 and so as the total number of participants(`qscore[k]`) who answer correctly for that question.

Another **IF statement** is used to calculate the **score of each school**:

```
begin  
    for j:=1 to nos do  
        if pschool[i]=school[j] then  
            sscore[j]:=sscore[j]+pscore[i];  
end;
```

After the scores of each participant have been calculated, these lines can be executed to calculate the score of each school. If the school name from the list of schools read is the same as the name of the school of a participant, the school score of that particular school will be incremented by the score of participant

One last procedure segment is used to calculate the mean score of all the participants.

```
for i:= 1 to n do  
    tscore:=tscore+pscore[i];
```

## Multiple-Choice Analysis Report

```
mean:=tscore/n;
```

The **total score of all participants ( tscore )** is first calculated by summing up all the score of participants. Then the mean is calculated by using the **formula**:

4. To sort the records  $\bar{X} = \frac{\sum X}{N}$

Before we can calculate the results and decide the award winners, we have to **rank the participants and schools**, therefore, procedures named `sorts` and `sortp` is written to **sort those respectively**.

Both sorting procedures adopt **insertion sort** as the algorithm. Although it is **much less efficient** on large lists than **more advanced algorithms** such as quicksort or merge sort, there are some **advantages** of insertion sort.

Firstly, **ease of implementation** is one of the main concerns of the programmer. As implementing insertion sort is far easier than those advanced algorithms, it will be adopted in this program.

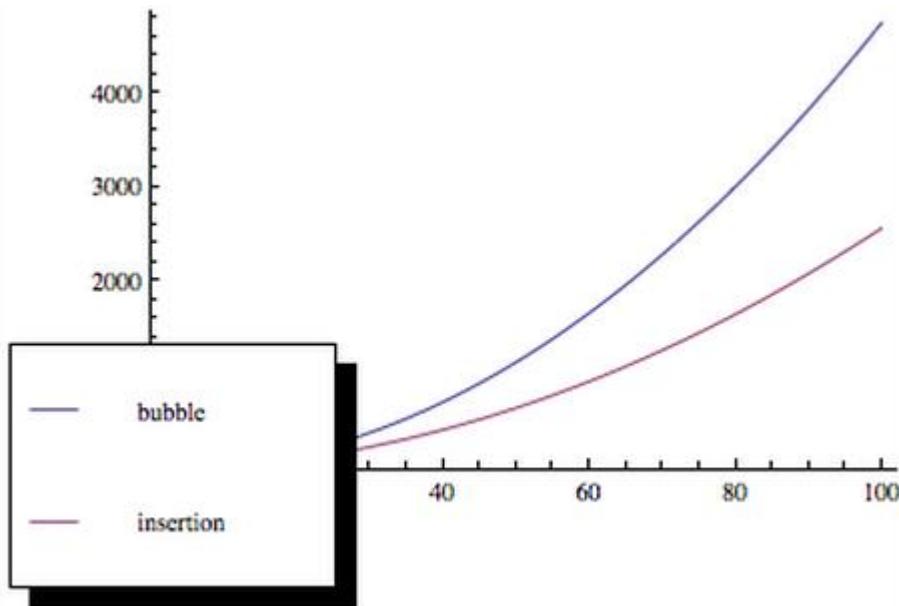
Secondly, it is **efficient enough for small data sets**. Compared to other quadratic sorting algorithms, like bubble sort, the number of swaps for elements in insertion sort is smaller if the list of data is nearly sorted.

Comparing their average number of comparisons:

- **Bubble sort:**  $\frac{1}{2} (N^2 - N \ln N - (\gamma + \ln 2 - 1)N) + \mathcal{O}(\sqrt{N})$
- **Insertion sort:**  $\frac{1}{4} (N^2 - N) + N - H_N$



## Multiple-Choice Analysis Report



We can see that using insertion sort will be substantially faster, so using insertion sort will be a **reasonable** choice.

A graphical representation of insertion sort:

**Step 1**

Assume first item is "sorted"

5	2	6	1	3	9
---	---	---	---	---	---

**Step 2**

Identify the value to compare

5		6	1	3	9
---	--	---	---	---	---

2
---

**Step 3**

Since  $5 > 2$ , shift 5 over to create space for 2 in the sorted section

	5	6	1	3	9
--	---	---	---	---	---

2
---

**Step 4**

Insert 2 into the empty space in the sorted section

2	5	6	1	3	9
---	---	---	---	---	---

In real implementation, there are **N-1 passes** in an insertion sort for a list of N elements. If pass denotes the number of passes, the position of the element to be inserted will be pass + 1, and the elements on the sorted list to be searched for will be from `pscore[1]` to `pscore[pass]`, `pname[1]` to `pname[pass]`, `sex[1]` to `sex[pass]`, `pschool[1]` to `pschool[pass]`. The information of participants will be sorted by the score they have in ascending order.

## Multiple-Choice Analysis Report

To **pick out** the element to be inserted, the followings are written:

```
elementscore := pscore[pass+1]; {pick out the elements  
elementname:=pname[pass+1]; to be inserted}  
elementschool:=pschool[pass+1];  
elementc:=sex[pass+1];
```

To search for the correct position to insert the picked elements, they are **compared with successive elements on the sorted list**:

```
position:=1;  
for i:= 1 to pass do  
  if elementscore>pscore[i]  
    then position:=i+1;
```

To **shift the elements** which are greater than the picked elements, swapping is done from the greatest element `pscore[pass+1]` on the sorted list:

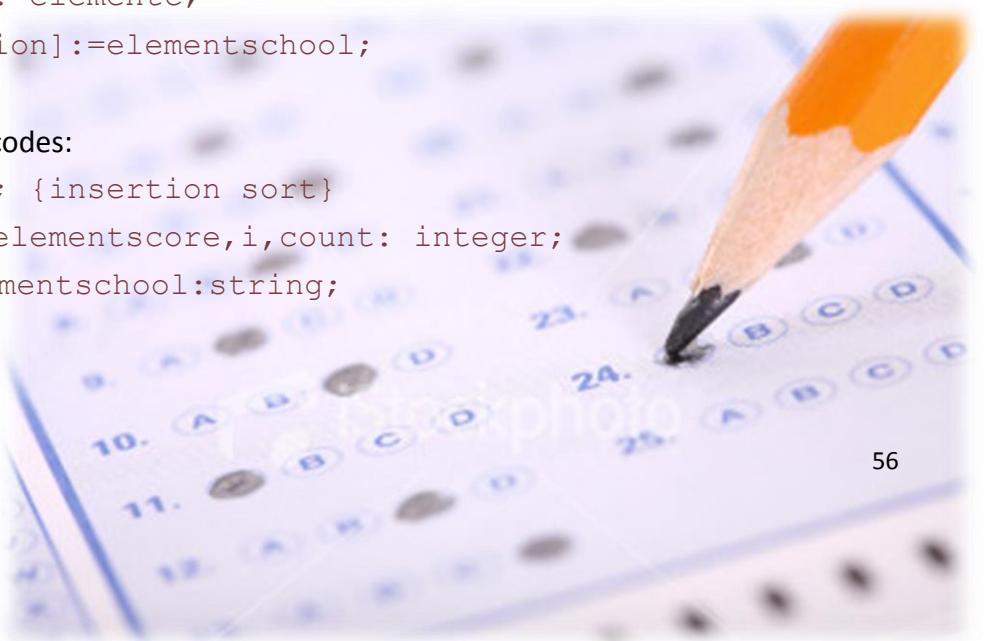
```
for count:= pass downto position do begin  
  pscore[count+1]:=pscore[count];  
  pname [count+1]:=pname [count];  
  sex [count+1]:=sex [count];  
  pschool [count+1]:=pschool [count];  
end;
```

After the elements have been shifted downwards, the picked elements can be **inserted in the correct positions**:

```
pscore[position]:=elementscore;  
pname [position]:=elementname;  
sex [position]:=elementc;  
pschool [position]:=elementschool;
```

Entire procedure program codes:

```
{4}procedure sortp; {insertion sort}  
Var pass,position,elementscore,i,count: integer;  
  elementname,elementschool:string;  
  elementc:char;  
begin
```



## Multiple-Choice Analysis Report

```
For pass := 1 to n-1 do
Begin
    elementscore := pscore[pass+1];
    elementname:=pname[pass+1];
    elementschool:=pschool[pass+1];
    elementc:=sex[pass+1];
    position:=1;
    for i:= 1 to pass do
        if elementscore>pscore[i]
            then position:=i+1;

    for count:= pass downto position do begin
        pscore[count+1]:=pscore[count];
        pname[count+1]:=pname[count];
        sex[count+1]:=sex[count];
        pschool[count+1]:=pschool[count];
    end;

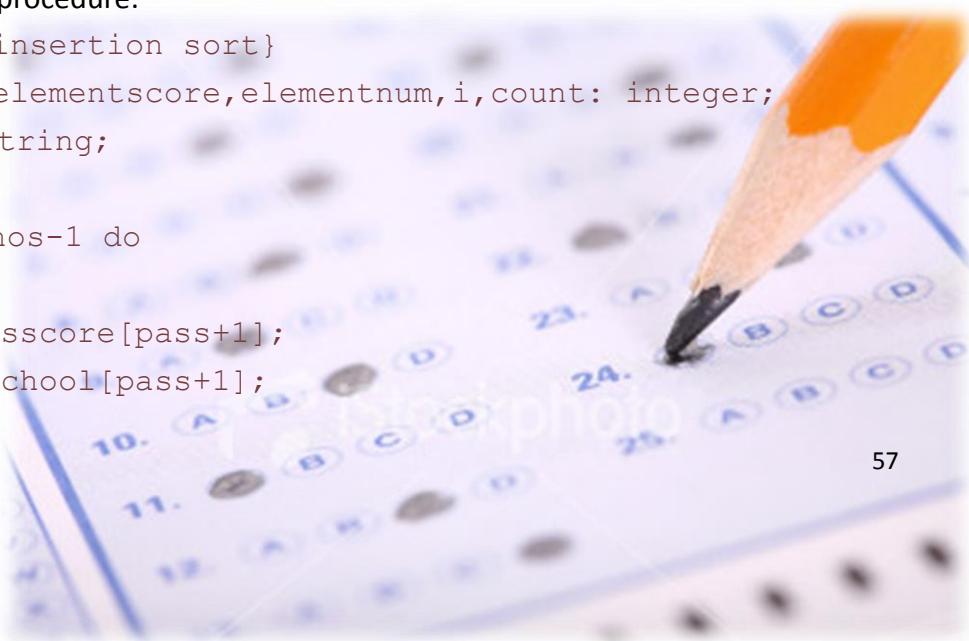
    pscore[position]:=elementscore;
    pname[position]:=elementname;
    sex[position]:=elementc;
    pschool[position]:=elementschool;

End;
end;
```

The **similar manner** is used to **sort the record of schools and their score( sorts)**, there are just a little bit of differences in the names of variables used and some data structures.

The program codes for the procedure:

```
procedure sorts; {insertion sort}
Var pass,position,elementscore,elementnum,i,count: integer;
    elementschool:string;
begin
    For pass := 1 to nos-1 do
        Begin
            elementscore := sscore[pass+1];
            elementschool:=school[pass+1];
```



## Multiple-Choice Analysis Report

```
elementnum:=snos [pass+1];
position:=1;
for i:= 1 to pass do
  if elementscore>sscore[i]
    then position:=i+1;

for count:= pass downto position do begin
  sscore[count+1]:=sscore[count];
  school[count+1]:=school[count];
  snos[count+1]:=snos[count];
end;

sscore[position]:=elementscore;
school[position]:=elementschool;
snos[position]:=elementnum;

End;
end;
```

## 5. Other functions

For other functions of the program,

- ⌚ To calculate the results of each school
- ⌚ To find out the award winners
- ⌚ To analyze questions and calculate for the statistics
- ⌚ To display results on the screen
- ⌚ To search for a particular participant
- ⌚ To save the report in text form for users

They are all written in the procedure `mainmenu`.

The main menu consists of **options** that the user can **choose to display some of the results**. The screenshots of this will be shown in the next section.

In order to **validate** the option inputted by the user, firstly, the variable storing the option will be of **string type** to **avoid program self-termination** when character/strings are being inputted in.

## Multiple-Choice Analysis Report

The procedure will make use of the built-in **val** function to check whether it is within the **range** of options available.

The user should only input an integer from **1 to 15**. The below program codes can show how it is done.

```
write('Please enter the option(1-15): ');
textcolor(15);
readln(option);
val(option,ioption,code);
lenoption:= length(option);
if  (code<>0) or (ioption<1) or (ioption>15)
then begin
    textcolor(3);
    gotoxy(27,c+2);
    writeln('Invalid input! Please try again.');
end;
c:=c+2;
end;
until (ioption<=15) and (ioption>0);
```

Lastly, only if the **numeric value** (ioption) is within the range from **1 to 15** that the program can **quit this REPEAT loop**.



## Multiple-Choice Analysis Report

The output of results on the screen will be first discussed, as proposed in [Chapter 3](#), the results will be:

- ⌚ Total number of participants, total number of participating schools and total number of participant(s) from each participating school
- ⌚ Winners of individual awards and school awards
- ⌚ Percentage correct for each question
- ⌚ Standard deviation of score in the whole competition.
- ⌚ Median / mode/mean score of in the whole competition.
- ⌚ Range, inter-quartile range of score
- ⌚ Score in upper quartile and lower quartile

### 1 . To display the numbers of participants and schools

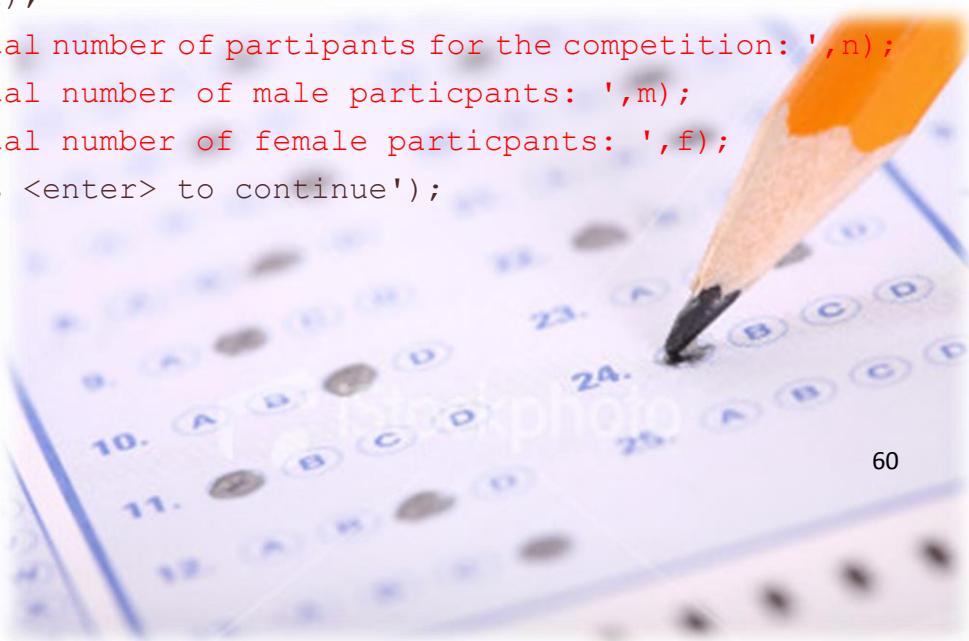
For displaying the numbers of participants and schools, several lines of program codes in the **CASE OF statement** are written. They are displaying,

- ⌚ the total number of participants (n)
- ⌚ total number of male participants (m)
- ⌚ total number of female participants (f)
- ⌚ and the total number of schools and their corresponding participants (nos)

For the **total number of participants, male participants** and **female participants**, the program will simply show the results got from counting the number of participants from the previous procedures.

Here are the program codes:

```
1: begin
    writeln;
    textcolor(11);
    writeln('Total number of partipants for the competition: ',n);
    writeln('Total number of male particpants: ',m);
    writeln('Total number of female particpants: ',f);
    write('Press <enter> to continue');
    readln;
    mainmenu;
end;
```



## Multiple-Choice Analysis Report

```
2: begin
    writeln;
    textcolor(11);
    writeln('Total number of schools for the competition: ',nos);
    write('Press <enter> to continue');
    readln;
    mainmenu;
end;
```

For the number of participants in each school, the program will display the **names of participating school** read from SCHOOL.TXT, then a **FOR loop** will be included in the program code to **display the names of schools** and their **numbers of participants** one by one.

```
for i:=1 to nos do begin
    gotoxy(5,j);
    writeln(school[i],snos[i]);
    j:=j+1;
end;
```



## 2 . To calculate and display the results and awards

For individual winners, as mentioned in [Chapter 2.1 Competition Regulations](#), participants are given awards **according to their percentiles**.

As percentiles indicate the **rank of a participant in the competition**, by **calculating their ranks** in the sorted list, the award winners can be easily shown.

For example, medal award winners are the ones who are in the 98<sup>th</sup>-100<sup>th</sup> percentile, therefore, by calculating the rank of 98-th percentile using this formula

$$n = \frac{P}{100} \times N + \frac{1}{2}$$

where n is the rank of participants, the list of participants who are medalists can be worked out.

In the program, firstly, the **rank** of participant in the 98<sup>th</sup> percentile is calculated,

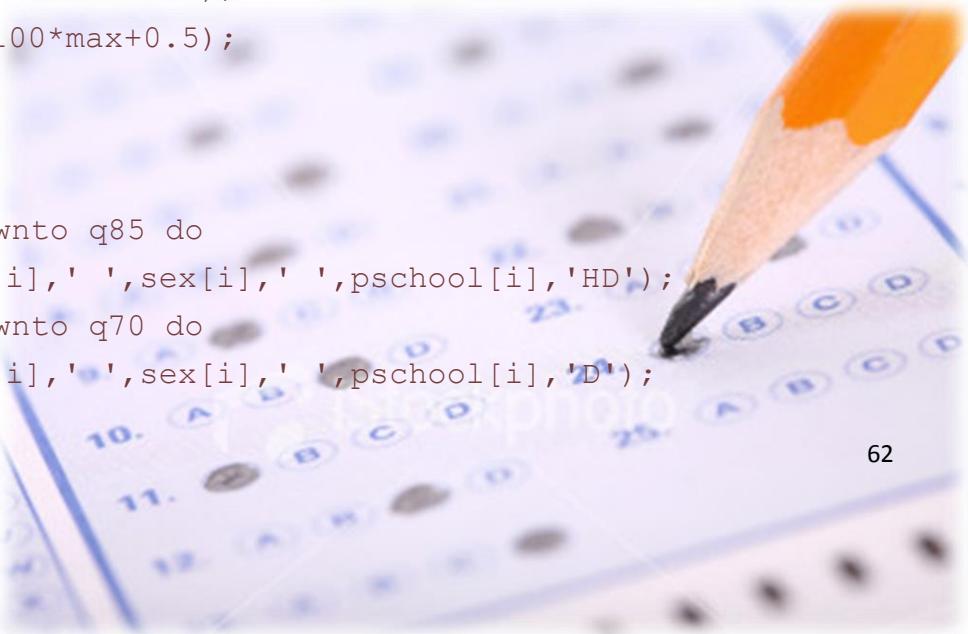
```
q98:=round(98/100*max+0.5);
```

Then, the **medalists will be shown one by one** with a **FOR loop**.

```
for i:=n downto q98 do  
    writeln(pname[i],' ',sex[i],' ',pschool[i],'Medal');
```

The **same method** is used to deal with the data of the other **awardees**, here are the program codes:

```
q85:=round(85/100*max+0.5);  
q70:=round(70/100*max+0.5);  
q50:=round(50/100*max+0.5);  
q35:=round(35/100*max+0.5);  
. . .  
for i:=q98-1 downto q85 do  
    writeln(pname[i],' ',sex[i],' ',pschool[i],'HD');  
for i:=q85-1 downto q70 do  
    writeln(pname[i],' ',sex[i],' ',pschool[i],'D');
```



## Multiple-Choice Analysis Report

```
for i:=q70-1 downto q50 do
    writeln(pname[i], ' ', sex[i], ' ', pschool[i], 'Merit');
for i:=q50-1 downto q35 do
    writeln(pname[i], ' ', sex[i], ' ', pschool[i], 'Credit');
writeln;
writeln('Key: HD=High Distinction, D=Distinction');
```

For **school award winners**, as the awards are given to winners according to their ranks, displaying the **list of award winners using a FOR loop** is fair enough.

```
writeln(school[nos], ' ', sscore[nos], ' Champion');
writeln(school[nos-1], ' ', sscore[nos-1], ' 1st Runner
Up');
writeln(school[nos-2], ' ', sscore[nos-2], ' 2nd Runner
Up');
for i:=nos-3 downto 1 do
    writeln(school[i], ' ', sscore[i], ' Participation');
```

### 3 . To analyze questions and calculate for the statistics

For analyzing the questions, one of the best ways is to look for the **percentage correct** of each question.

The percentage correct for a question can be calculated by **diving** the **total number of correct answers for a particular question** by the **total number of participants**, the percentage of i question can be calculated by such program line:

```
percentcorrect[i]:=qscore[i]/n*100;
```

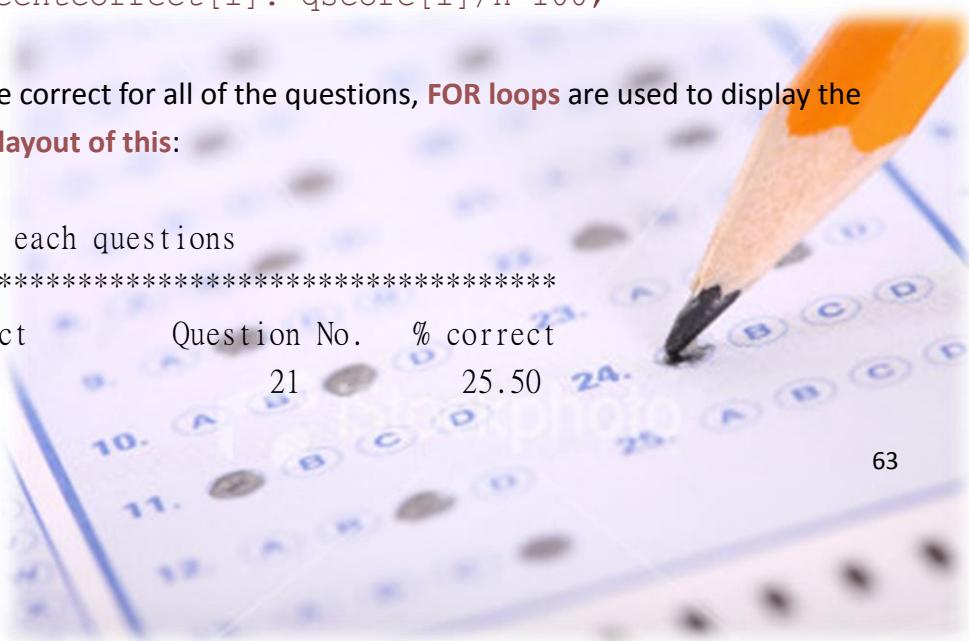
So to display the percentage correct for all of the questions, **FOR loops** are used to display the results on the screen like a **layout of this**:

Percentage correct for each questions

\*\*\*\*\*

Question No.	% correct
1	31.50

Question No.	% correct
21	25.50

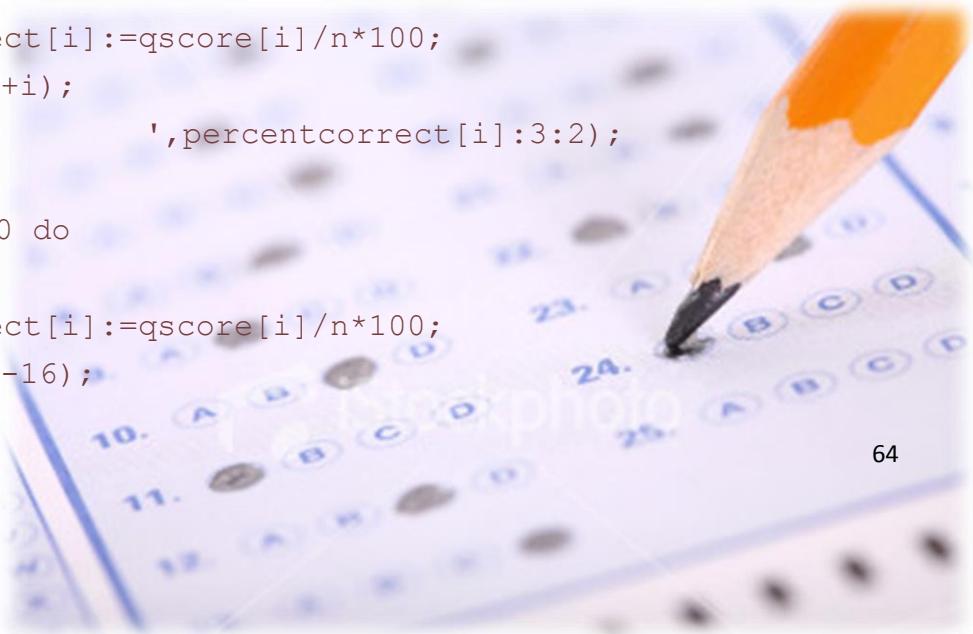


## Multiple-Choice Analysis Report

2	27.00	22	24.00
3	22.50	23	29.50
4	23.50	24	34.00
5	23.50	25	23.50
6	24.50	26	22.50
7	23.50	27	26.50
8	32.00	28	29.00
9	25.00	29	20.50
10	25.50	30	22.00
11	25.50	31	23.00
12	25.50	32	30.50
13	27.00	33	23.50
14	23.50	34	21.50
15	26.50	35	28.50
16	28.00	36	29.00
17	24.50	37	26.00
18	25.50	38	25.00
19	29.50	39	26.00
20	29.00	40	29.50

The program codes will be:

```
for i:= 1 to 9 do           { %correct }
begin
percentcorrect[i]:=qscore[i]/n*100;
gotoxy(21,4+i);
writeln(i,'      ',percentcorrect[i]:3:2);
end;
for i:= 10 to 20 do
begin
percentcorrect[i]:=qscore[i]/n*100;
gotoxy(21,4+i);
writeln(i,'      ',percentcorrect[i]:3:2);
end;
for i:=21 to 40 do
begin
percentcorrect[i]:=qscore[i]/n*100;
gotoxy(54,i-16);
```



## Multiple-Choice Analysis Report

```
writeln(i, ' ', percentcorrect[i]:3:2);  
end;
```

As some spacing differences are there for different lines, so there are differences for loops to alter the spacing between results to be shown.



## Multiple-Choice Analysis Report

Calculating standard deviation is an **effective method** to see whether the score of participants is dispersed or not. Therefore, standard deviation calculation is provided by the program.

To calculate the standard deviation of the score of participants, the **formula** of standard deviation should be considered.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

As standard deviation is given by the square root of the quotient when dividing sum of the square of difference between a datum and the mean of the set of data by total number of data, by using a FOR loop, **the sum of the square of difference between a datum and the mean of the set of data** can be calculated first.

First, we can initialize the real type variable `d` storing the sum first.

```
d:=0;
```

Then, **summing up the square of difference** between **a participant's score** and **the mean** of the set of scores:

```
for i:=1 to n do  
    d:=d+ (pscore[n]-mean) * (pscore[n]-mean);
```

Finally the standard deviation (`SD`) is simply **the square root of the quotient** when it is **divided by the total number of participants**,

```
SD:=sqrt(d/n);
```

Then, the standard deviation will be **displayed**.

```
writeln('Standard deviation of score is: ', SD:3:2);
```

The **mean** is already calculated in previous procedure `calscore`, therefore it can easily be displayed by:

```
writeln('Mean: ', mean:3:2);
```

while the **median** is given by the score at the middle of the rank:

```
writeln('Median: ', pscore[round((1+max)/2)]);
```

## Multiple-Choice Analysis Report

The **range** of score is given by the difference between the lowest score and the highest score. As the array of score of participants has been sorted in ascending order according to their scores, we can simply get the range by calculating the difference between `pscore[n]` and `pscore[1]`:

```
writeln('Range: ', pscore[n]-pscore[1]);
```

The **upper and lower quartile** of score of participants are calculated in a similar manner like what is done to deciding the individual award winners, the only difference is that the upper quartile is the 75-th quartile while the lower quartile is the 25-th quartile.

```
writeln('Upper quartile :', pscore[round(75/100*n+0.5)]);
writeln('Lower quartile :', pscore[round(25/100*n+0.5)]);
```

And the **interquartile range** is the difference of these two:

```
writeln('Inter-quartile
range :', pscore[round(75/100*n+0.5)]-pscore[round(25/100*n+0.5)])
; 
```

## 4 . To search for a particular participant

Search functions are also included in the program, by 3 ways, namely, **by name**, **by school** and **by sex**.

The user can first **choose** searching participant **by what way** in the main menu and then they can input the designated name (`targetname`)/ school (`targetschool`)/ sex (`targetsex`).

For searching by name, firstly, to **avoid the program misunderstanding** a lower case version of the name inputted is a wrong input, all the characters in the string stored will be **turned to upper cases first**.

```
readln(targetname);
  for i:=1 to length(targetname) do
    targetname[i]:=upcase(targetname[i]);
```

The same for school:

```
readln(targetschool);
  for i:=1 to length(targetschool) do
```

## Multiple-Choice Analysis Report

```
targetname[i]:=upcase(targetschool[i]);
```

Yet, as the defined data structure for the names and school names of participants are arrays of strings that have a **definite size** ( 25 and 35). When user is inputting the name/name of school, **empty spaces** are needed to be added to the string storing the target so that a search can be successfully done.

For example, for the name of participant, the data input by user will be stored into **a string** targetname, and then spaces will be added in a **FOR loop**.

```
for i:=1 to (25-length(targetname)) do  
    targetname:= targetname+' ';
```

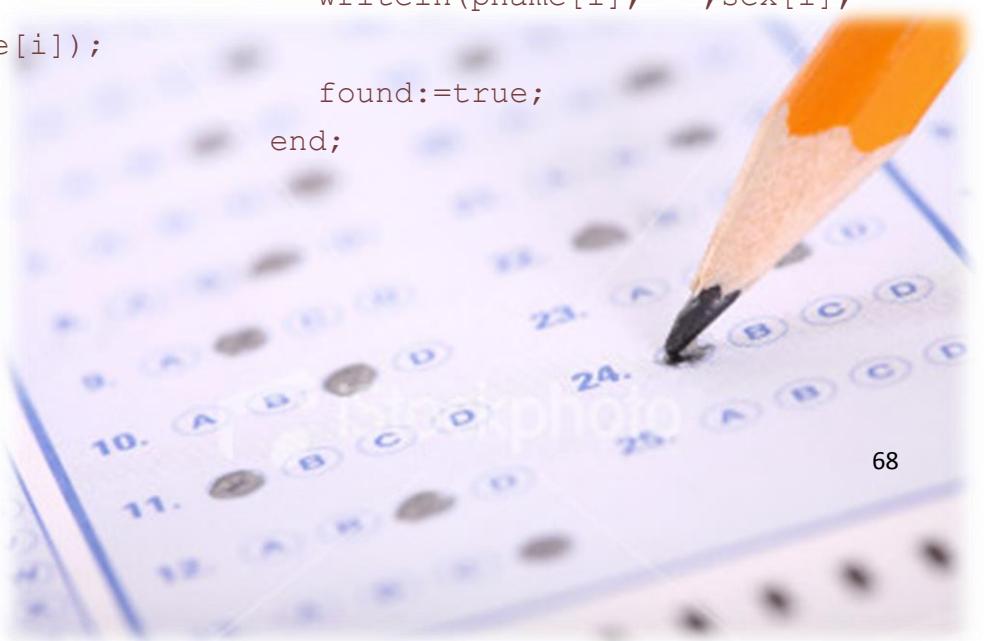
The same for school:

```
for i:=1 to (45-length(targetschool)) do  
    targetschool:= targetschool+' ';  
    writeln(length(targetschool));
```

As for **searching algorithm**, **linear search** is used. This is because, linear search, comparing to binary search, have several **advantages**. The primary advantage of linear search is its simplicity: conceptually, it is extraordinarily **easy to understand**, and, **implementation-wise**, it is also very **straight-forward**. Even though its executional speed will increase with large data size, it will be **fast enough** to search for results in a **small size** of 200 participants.

the whole array of name of participant is scanned through until a name in the array is equivalent to that of the target name. A **FOR loop** is used to solve this and display the record.

```
for i:=n downto 1 do  
    if targetname=pname[i] then begin  
        writeln(pname[i], ' ', sex[i], '  
' , pschool[i], pscore[i]);  
        found:=true;  
    end;
```



## Multiple-Choice Analysis Report

Similarly, for the searching by school and by sex, the records can be displayed like this:

```
for i:=n downto 1 do
    if targetschool=pschool[i] then begin
        writeln(pname[i], ' ', sex[i], '
', pschool[i], pscore[i]);
        found:=true;
    end;

for i:=n downto 1 do
    if targetsex=sex[i]
        then writeln(pname[i], ' ', sex[i], '
', pschool[i], pscore[i]);
```

If no matched record is found, a 'RECORD NOT FOUND' statement will be displayed:

```
if found=false then writeln('RECORD NOT FOUND');
```

where `found` is a **flag** that shows whether a record has matched the target.



## Multiple-Choice Analysis Report

As stated in [Chapter 3](#), the analysis report can also outputted as a text file format, so that the report can be more **portable**.

The outputting method to a text file is actually the same when we display results on the screen, but with the **file opened** in the program for **writing**.

The user will first be asked to input the file name they want. The file name will be stored in a **string called** (`textfile`).

```
write('Enter the name of the text file you want to save as  
(with .txt): ');\nreadln(textfile);
```

A **simulation** to the **program loading** is included in the program to make the users **convinced** that the program is **really producing a file**. This feature is added because otherwise the making of file will be **too fast** that the user **may not have realized it**.

```
write('Loading');\ndelay(1000);\nwrite('.');\ndelay(1000);\nwrite('.');\ndelay(1000);\nwrite('.');\nwriteln('Success');
```

The file will then be assigned to a text type variable called `outfile` for writing.

The program codes for this section:

```
assign(outfile,textfile);\nrewrite(outfile);\nwriteln(outfile,'A Report on the ICT competition'); //title\nwriteln(outfile,'-----');\nwriteln(outfile);\nwriteln(outfile,'Participation: ');\nwriteln(outfile,'*****'); //No. of participants\nwriteln(outfile,'Total number of participants for the competition: ',n);
```

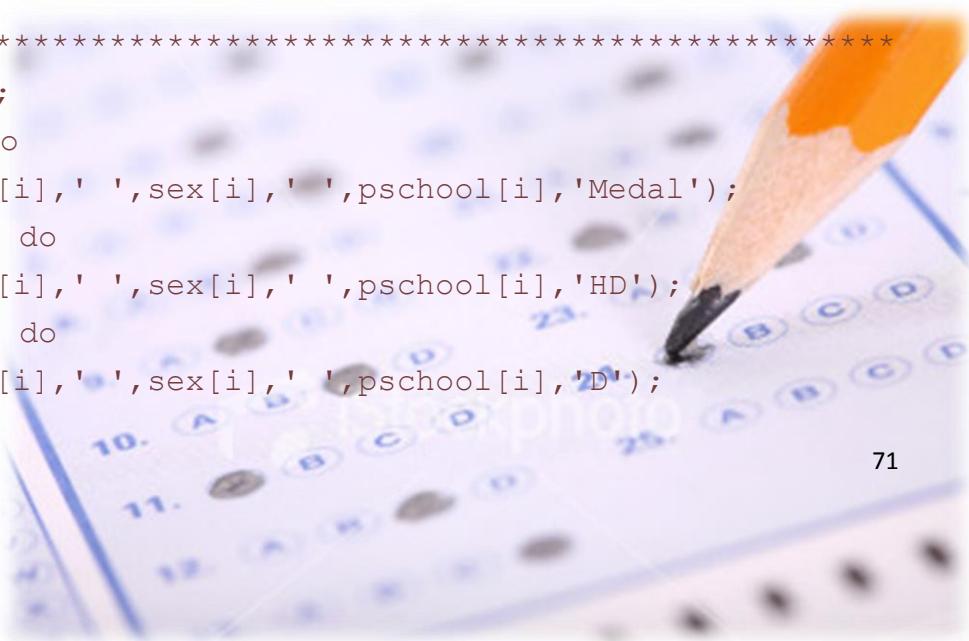
## Multiple-Choice Analysis Report

```
writeln(outfile,'Total number of male participants: ',m);
writeln(outfile,'Total number of female participants: ',f);
writeln(outfile,'Total number of schools for the competition: ',nos);
writeln(outfile);
writeln(outfile);
writeln(outfile,'Total number of participants from each schools');
writeln(outfile,'*****');
writeln(outfile,'School          No. of
participants');

writeln(outfile,'*****');
for i:=1 to nos do begin
    writeln(outfile,school[i],snos[i]); //Awards determination
    end;
writeln(outfile);
writeln(outfile);
q98:=round(98/100*max+0.5);
q85:=round(85/100*max+0.5);
q70:=round(70/100*max+0.5);
q50:=round(50/100*max+0.5);
q35:=round(35/100*max+0.5);

writeln(outfile,'AWARDS:');
writeln(outfile);
writeln(outfile,'Individual awardees of the competition');
writeln(outfile,'*****');
writeln(outfile,'Name          Sex  School
Award');

writeln(outfile,'*****');
for i:=max downto q98 do
    writeln(outfile,pname[i],' ',sex[i],' ',pschool[i],'Medal');
for i:=q98-1 downto q85 do
    writeln(outfile,pname[i],' ',sex[i],' ',pschool[i],'HD');
for i:=q85-1 downto q70 do
    writeln(outfile,pname[i],' ',sex[i],' ',pschool[i],'D');
```

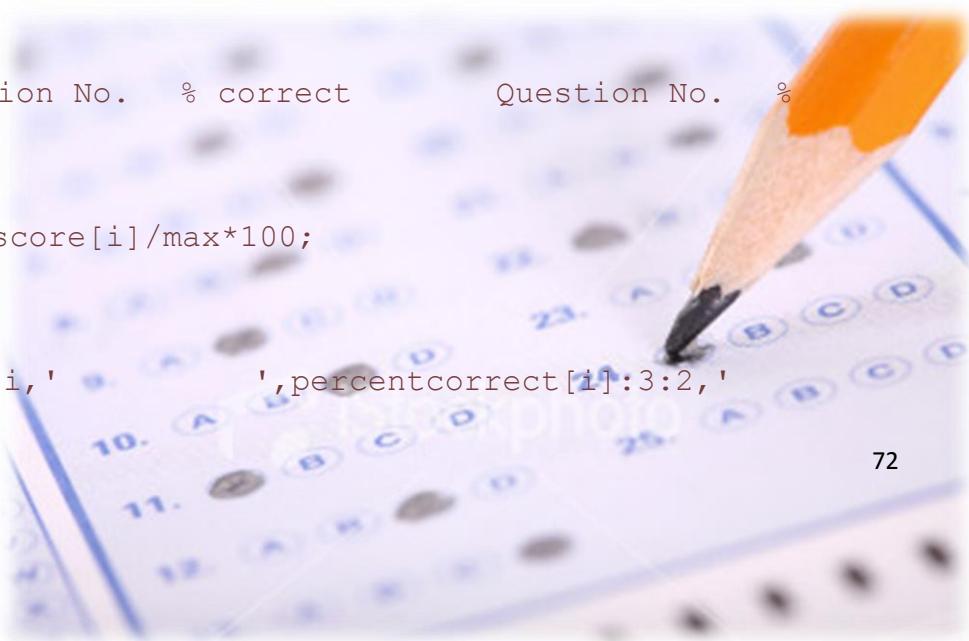


## Multiple-Choice Analysis Report

```
for i:=q70-1 downto q50 do
  writeln(outfile,pname[i],' ',sex[i],' ',pschool[i],'Merit');
for i:=q50-1 downto q35 do
  writeln(outfile,pname[i],' ',sex[i],' ',pschool[i],'Credit');
  writeln;
writeln(outfile,'Key: HD=High Distinction, D=Distinction');
writeln(outfile);
writeln(outfile);
writeln(outfile,'School award winners of the competition');
writeln(outfile,'*****');
writeln(outfile,'School Award' Total sum of score
Award');

writeln(outfile,'*****');
writeln(outfile,school[nos],' ',sscore[nos],' Champion');
writeln(outfile,school[nos-1],' ',sscore[nos-1],' 1st
Runner Up');
writeln(outfile,school[nos-2],' ',sscore[nos-2],' 2nd
Runner Up');
for i:=nos-3 downto 1 do
  writeln(outfile,school[i],' ',sscore[i],' Participation');
  writeln(outfile);
writeln(outfile);
writeln(outfile,'QUESTIONS AND SCORE ANALYSIS:');
writeln(outfile); // %correct
writeln(outfile,'Percentage correct for each questions');

writeln(outfile,'*****');
writeln(outfile,'Question No. % correct Question No. %
correct');
for i:=1 to 40 do
  percentcorrect[i]:=qscore[i]/max*100;
  i:=1;
  for i:=1 to 9 do
    writeln(outfile,' ',i,' ',percentcorrect[i]:3:2,'
```



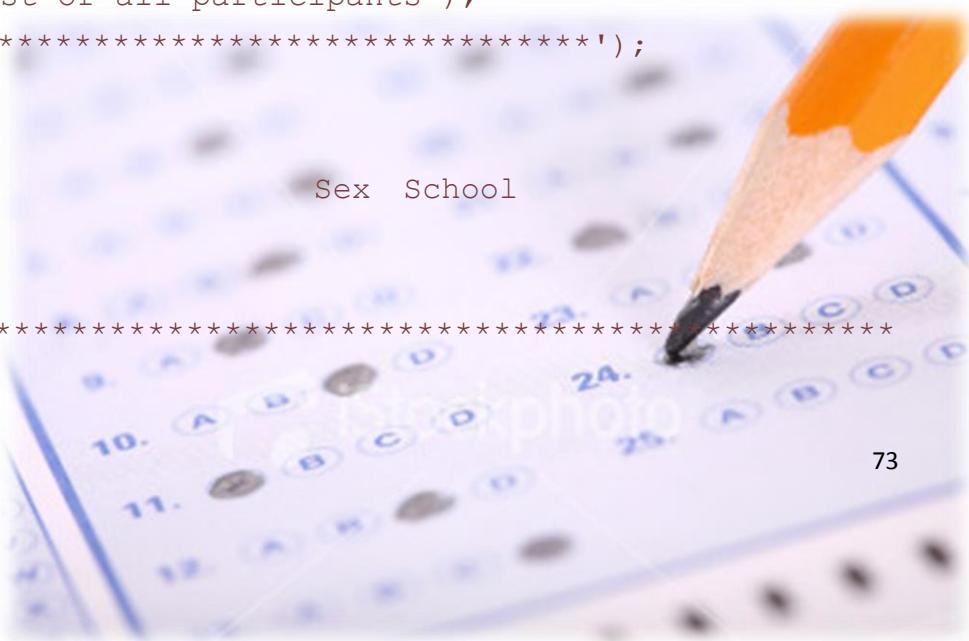
## Multiple-Choice Analysis Report

```
',i+20,'      ',percentcorrect[i+20]:3:2);
for i:=10 to 20 do
writeln(outfile,'    ',i,'      ',percentcorrect[i]:3:2,
',i+20,'      ',percentcorrect[i+20]:3:2);

d:=0;
for i:=1 to n do
d:=d+ (pscore[n]-mean)*(pscore[n]-mean);
SD:=sqrt(d/n);           //Standard deviation
writeln(outfile);
writeln(outfile);
writeln(outfile,'Standard deviation of score is: ',SD:3:2);
writeln(outfile,'Mean: ',mean:3:2);
writeln(outfile,'Median: ',pscore[round((1+max)/2)]); //Median
writeln(outfile,'Range: ',pscore[n]-pscore[1]); //Range
writeln(outfile,'Upper quartile :',pscore[round(75/100*n+0.5)]);
//Upper quartile
writeln(outfile,'Lower
quartile :',pscore[round(25/100*n+0.5)]); //Lower quartile
writeln(outfile,'Inter-quartile
range :',pscore[round(75/100*n+0.5)]-pscore[round(25/100*n+0.5)]); //
Inter-quartile range
writeln(outfile);
writeln(outfile);
writeln(outfile);
writeln(outfile); // Listing all participants
writeln(outfile,'The list of all participants');
writeln(outfile,'*****');
writeln(outfile);

writeln(outfile,'Name          Sex  School
Score');

writeln(outfile,'*****');
writeln(outfile,'*****');
```



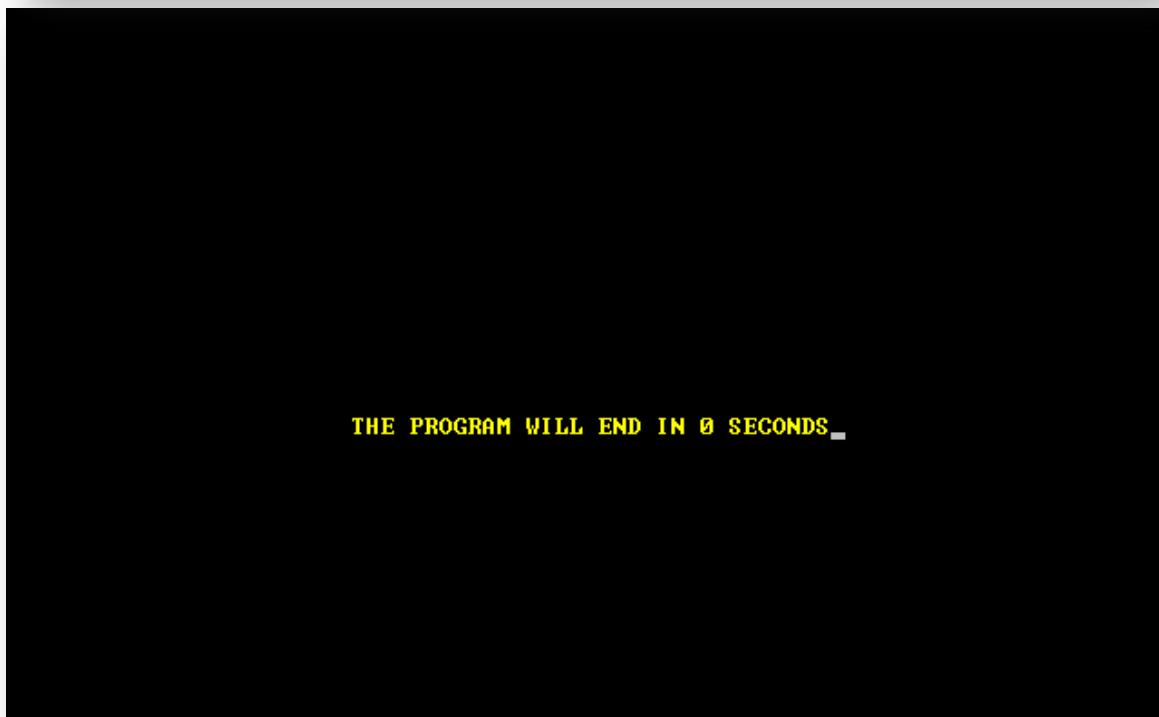
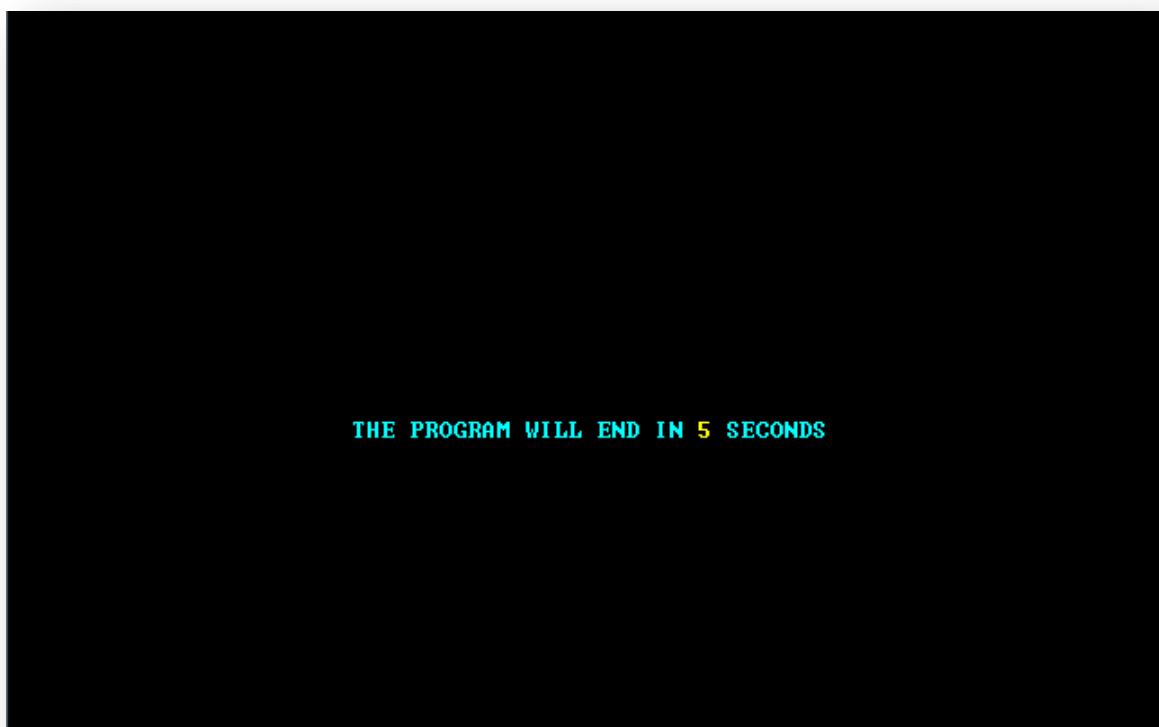
## Multiple-Choice Analysis Report

```
for i:=max downto 1 do  
writeln(outfile,pname[i],' ',sex[i],' ',pschool[i],pscore[i]);
```

## 6. Program termination

If the user chooses to quit the program, an **exit screen** will be shown to the user.

This makes use of the functions from the **crt library** to **restrict the word display area** to the centre of the screen and to make **colour variations** for showing a “flash” effect.



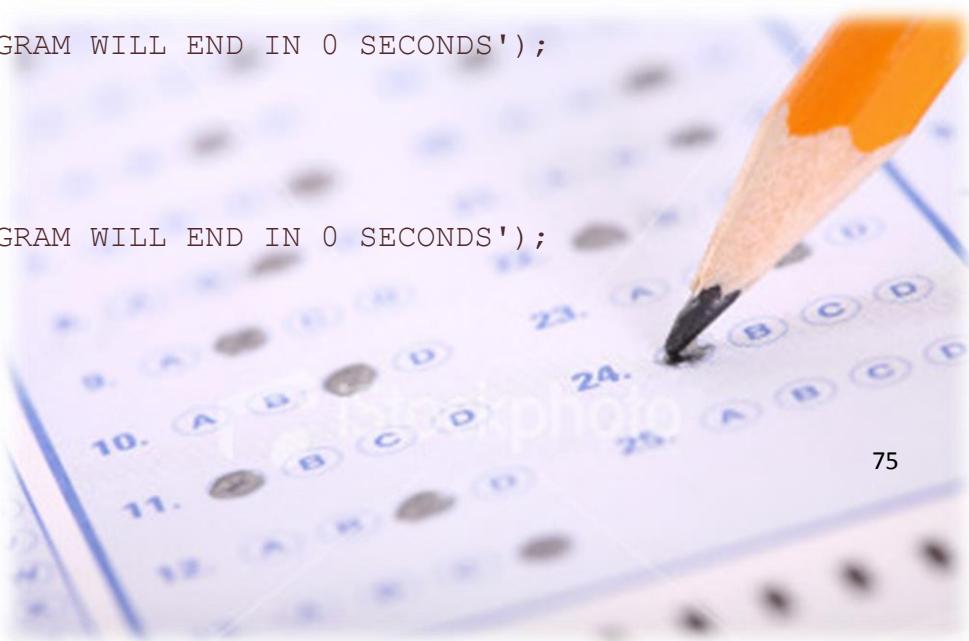
## Multiple-Choice Analysis Report

This makes the program **not so boring** by adding some **vibrant features** to it.

Here are the program codes:

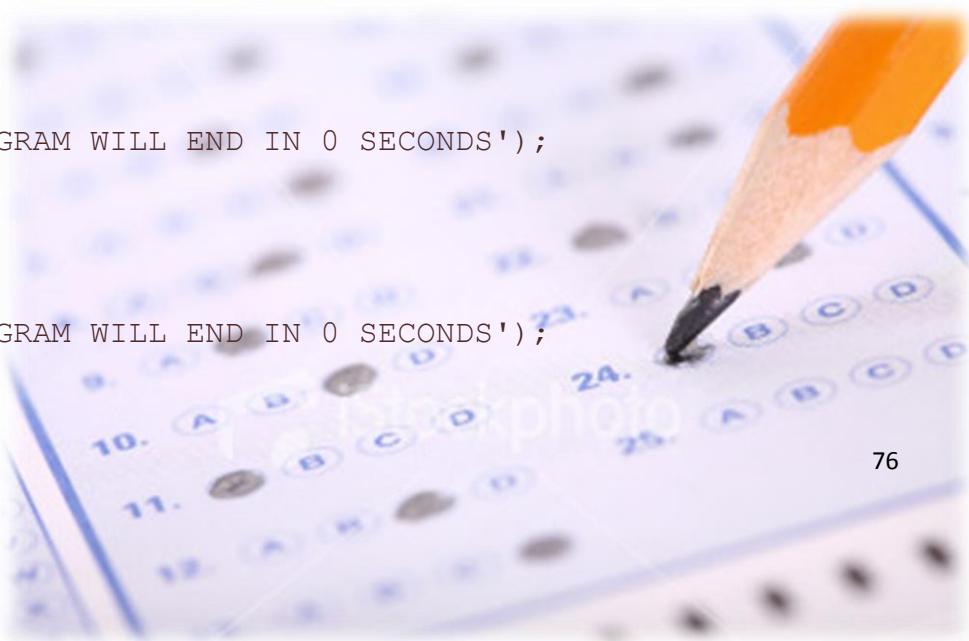
```
write('THE PROGRAM WILL END IN    SECONDS');
      for i:=5 downto 1 do
        begin
          textColor(14);
          gotoxy(49,15);
          write(i);
          delay(1000);
        end;

        gotoxy(25,15);
        textColor(14);
        write('THE PROGRAM WILL END IN 0 SECONDS');
        delay(100);
        gotoxy(25,15);
        textColor(11);
        write('THE PROGRAM WILL END IN 0 SECONDS');
        delay(100);
        gotoxy(25,15);
        textColor(14);
        write('THE PROGRAM WILL END IN 0 SECONDS');
        delay(100);
        gotoxy(25,15);
        textColor(11);
        write('THE PROGRAM WILL END IN 0 SECONDS');
        delay(100);
        gotoxy(25,15);
        textColor(14);
        write('THE PROGRAM WILL END IN 0 SECONDS');
        delay(100);
        gotoxy(25,15);
        textColor(11);
        write('THE PROGRAM WILL END IN 0 SECONDS');
        delay(100);
        gotoxy(25,15);
        textColor(14);
        write('THE PROGRAM WILL END IN 0 SECONDS');
```



## Multiple-Choice Analysis Report

```
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(11);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(14);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(11);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(14);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(11);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(14);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(11);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(14);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(11);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(14);
write('THE PROGRAM WILL END IN 0 SECONDS');
```



## Multiple-Choice Analysis Report

```
gotoxy(25,15);
textcolor(14);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(11);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
halt;
```

## 4.3 Data Structures

The below table shows a collection of **constants**, **variables** and **data structures** used in the program.

### Constants:

Name	Value	Description
max	200	Stores the maximum number of participants

### Variables:

#### Main program:

#### Participants

Name	Type (parallel arrays)	Description
pname	array[1..max] of string[25]	Storing the name of a particular participant
pscore	array[1..max] of integer	Storing the score of a particular participant
pans	array[1..max] of string[40]	Storing the answers for the questions of a particular participant
sex	array[1..max] of char	Storing the sex of a particular participant
pschool	array[1..max] of string[45]	Storing the names of school of a particular participant

## Multiple-Choice Analysis Report

### Schools

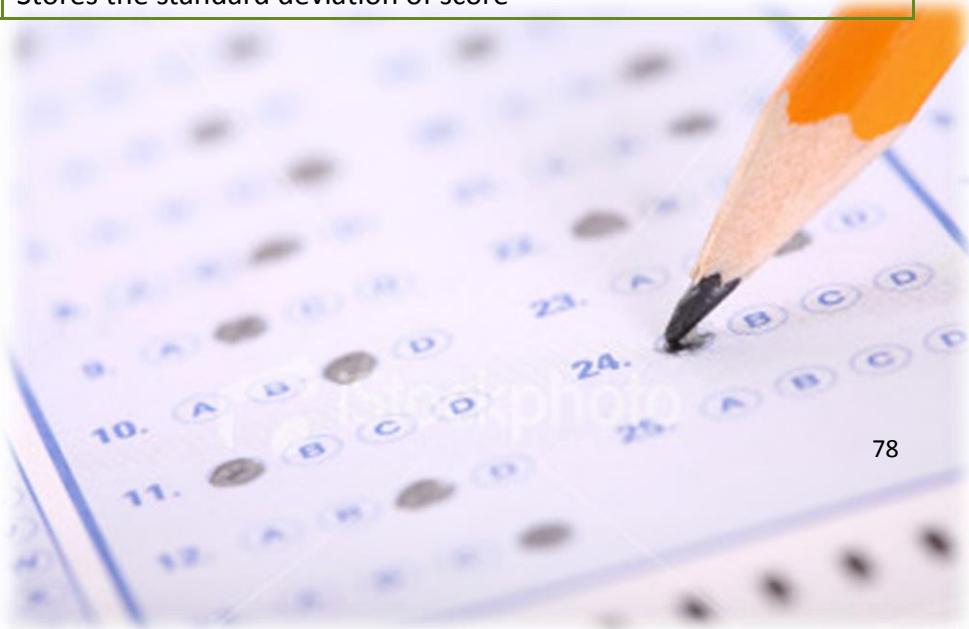
Name	Type(parallel arrays)	Description
<b>snos</b>	array[1..999] of integer	Stores the total number of participants in each participating school
<b>sscore</b>	array[1..999] of integer	Stores the score of each participating school
<b>school1</b>	array[1..999] of string[45]	Stores the names of participating schools

### Questions

Name	Type(parallel arrays)	Description
<b>qscore</b>	array[1..40] of integer	Stores the number of participants who answered correctly for a particular question
<b>percentcorrect</b>	array[1..40] of real	Stores the percentage correct of a particular question

Other variables:

Name	Type	Description
<b>n</b>	integer	Stores total number of participants
<b>m</b>	integer	Stores total number of male participants
<b>f</b>	integer	Stores total number of female participants
<b>nos</b>	integer	Stores total number of participating schools
<b>tscore</b>	integer	Stores total sum of scores of all participants
<b>ans</b>	string	Stores the file name of the answer key file
<b>schoollist</b>	string	Stores the file name of the list of school file
<b>mean</b>	real	Stores the mean of score of the competition
<b>median</b>	real	Stores the median of score
<b>d</b>	real	Stores the sum of difference between a score and the mean of score in the calculation of standard deviation
<b>SD</b>	real	Stores the standard deviation of score



## Procedures:

readinput

Name	Type	Description
<b>infile</b>	text	Stores the file to be inputted by the user
<b>pinfo</b>	string	Stores the name inputted of text file of the participants' info or answer key
<b>line</b>	string	Stores a line of text read from the inputted file
<b>flag</b>	boolean	Determines whether the name of file inputted is accepted

readschool

Name	Type	Description
<b>infile</b>	text	Stores the file to be inputted by the user
<b>line</b>	string	Stores a line of text read from the inputted file
<b>i</b>	integer	A program counter for pschool[i]
<b>j</b>	integer	A program counter for school[j]
<b>flag</b>	boolean	Determines whether the name of file inputted is accepted

calscore

Name	Type	Description
<b>i</b>	integer	A program counter for the FOR loop
<b>j</b>	integer	A program counter for the FOR loop
<b>k</b>	integer	A program counter for the FOR loop



## Multiple-Choice Analysis Report

sortp

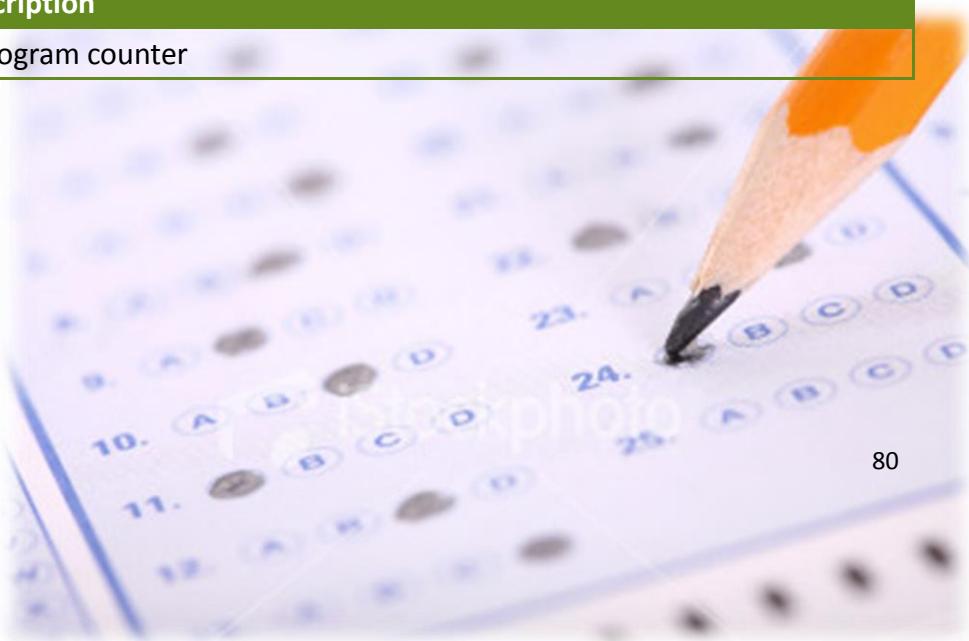
Name	Type	Description
<b>pass</b>	integer	Stores the number of passes for the insertion sort
<b>position</b>	integer	Stores the position of the last element in the sorted list
<b>i</b>	integer	A program counter to search for the correct position to insert the picked elements
<b>count</b>	integer	A program counter to shift the elements which are greater than the picked elements
<b>elementscore</b>	integer	A temporary for the score of each participant to be inserted
<b>elementname</b>	string	A temporary for the name of each participant to be inserted
<b>elementschool</b>	string	A temporary for the school name of each participant to be inserted
<b>elementc</b>	char	A temporary for the sex of each participant to be inserted

sorts

Name	Type	Description
<b>pass</b>	integer	Stores the number of passes for the insertion sort
<b>position</b>	integer	Stores the position of the last element in the sorted list
<b>i</b>	integer	A program counter to search for the correct position to insert the picked elements
<b>count</b>	integer	A program counter to shift the elements which are greater than the picked elements
<b>elementscore</b>	integer	A temporary for the score of each school to be inserted
<b>elementnum</b>	string	A temporary for the number of participants of each school to be inserted
<b>elementschool</b>	string	A temporary for the name of each school to be inserted

Readdisplay

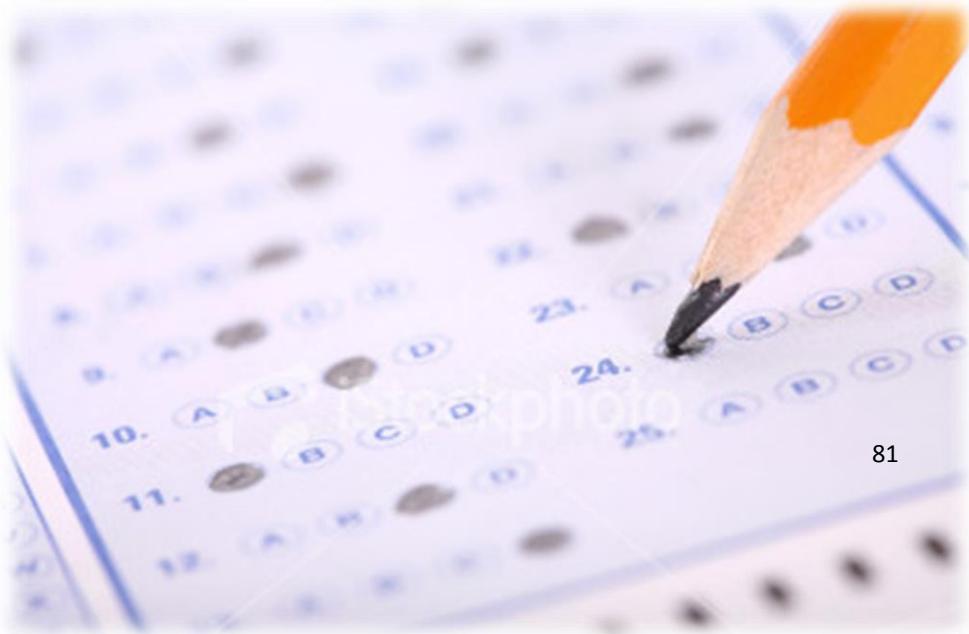
Name	Type	Description
<b>c</b>	integer	A program counter



## Multiple-Choice Analysis Report

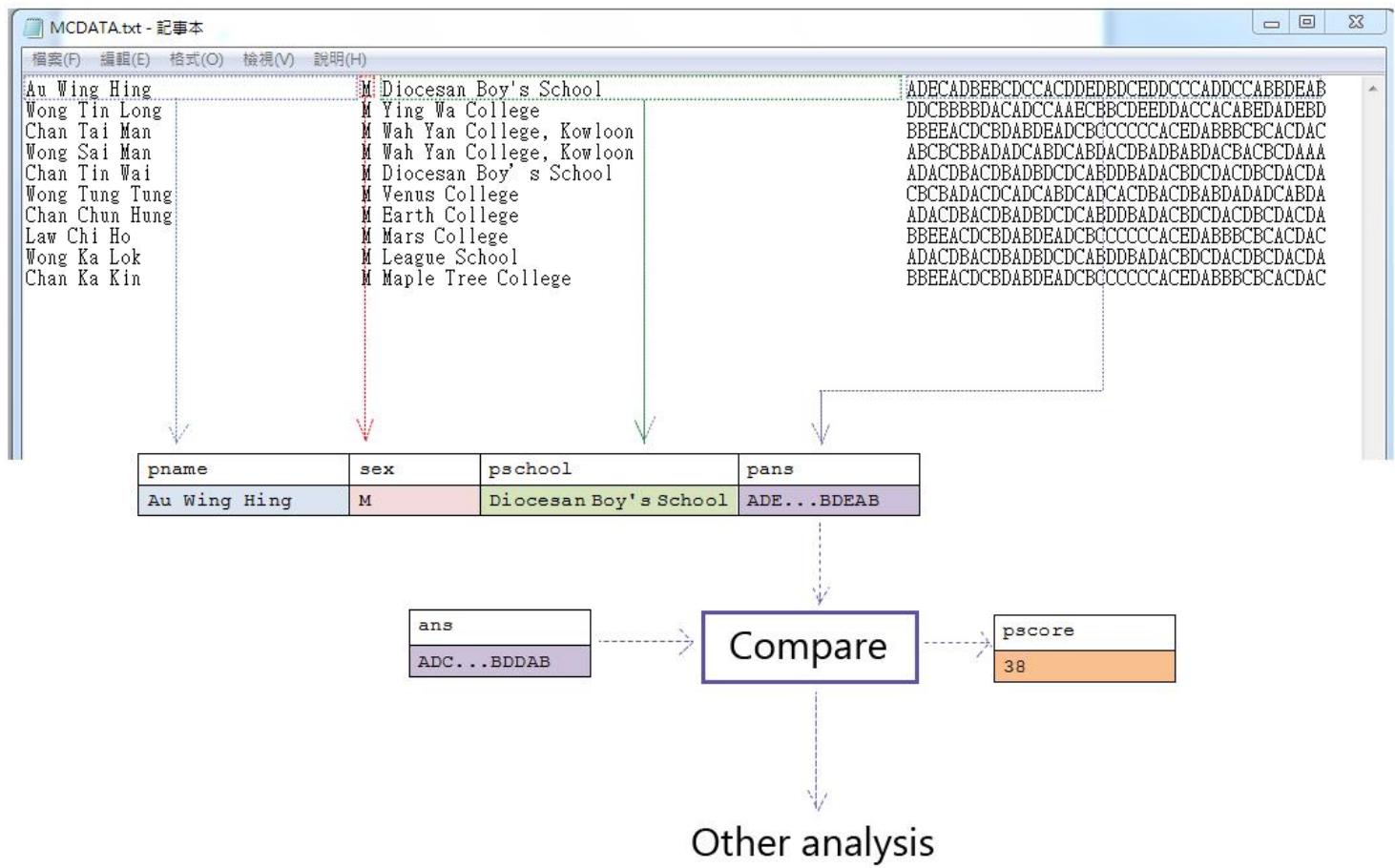
mainmenu

Name	Type	Description
<b>q98</b>	integer	Stores the position of the 98-th percentile of a participant in the array with ascending score of participant
<b>Q85</b>	integer	Stores the position of the 85-th percentile of a participant in the array with ascending score of participant
<b>Q70</b>	integer	Stores the position of the 70-th percentile of a participant in the array with ascending score of participant
<b>Q50</b>	integer	Stores the position of the 50-th percentile of a participant in the array with ascending score of participant
<b>Q35</b>	integer	Stores the position of the 35-th percentile of a participant in the array with ascending score of participant
<b>c</b>	integer	A program counter
<b>i</b>	integer	A program counter
<b>j</b>	integer	A program counter
<b>code</b>	integer	Stores the return value if the <code>val</code> function fails to be executed
<b>ioption</b>	integer	Stores the integer value of the numeric string inputted
<b>lenoption</b>	integer	Stores the length of inputted string
<b>targetname</b>	string	Stores the name of participant to be searched
<b>targetschool</b>	string	Stores the name of school to be searched
<b>textfile</b>	string	Stores the file name of the analysis report to be outputted
<b>option</b>	string	Stores the
<b>targetsex</b>	char	Stores the sex of participant to be searched
<b>outfile</b>	text	Stores the file to be outputted by the user
<b>found</b>	boolean	Determines whether the inputted information is found in the list of participants



## Multiple-Choice Analysis Report

The following diagram can summarise the program data flow:



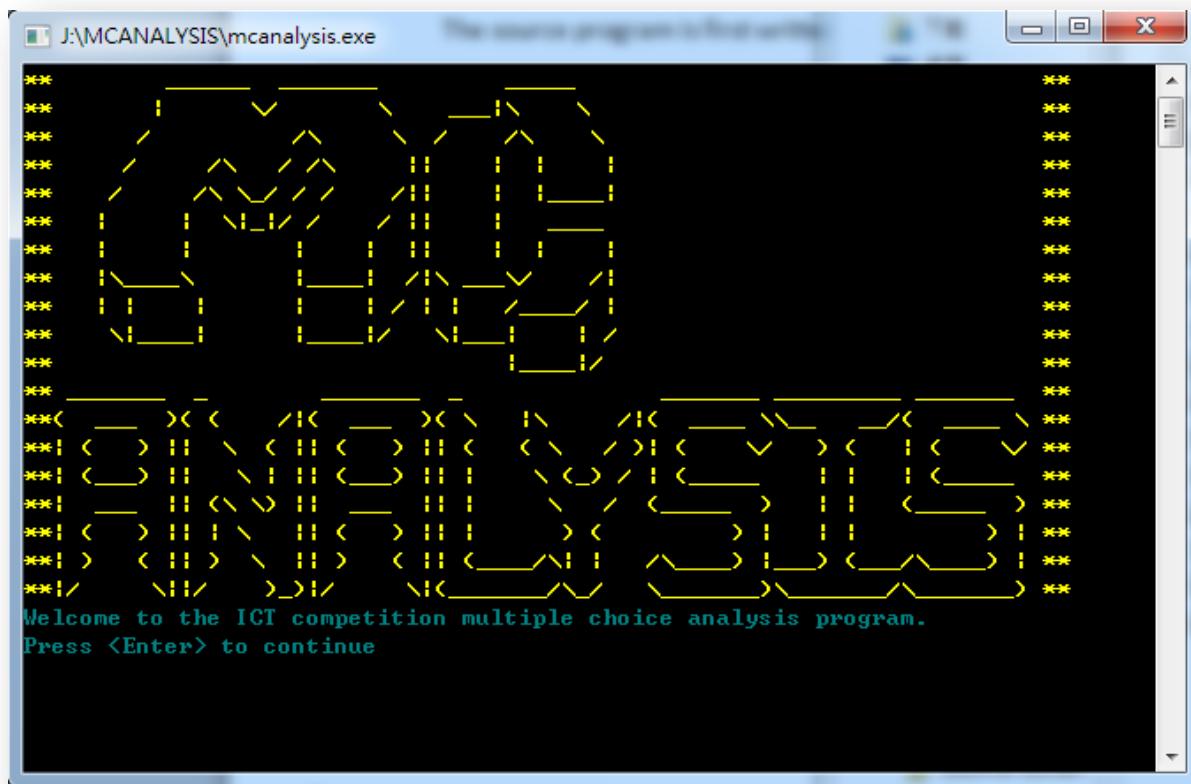
## 4.4 Program Coding

Program coding is done by using **Pascal** as the programming language. The **compiler**, as stated in [Chapter 2](#), is Dev-Pascal. As seen from the below screenshot, Dev-pascal provides a **clearly structured** user-interface, and different **colours** are used to identify **different identifiers**, which is super **helpful** to the programmer.

The source program is first written in the Pascal compiler:

The file name of the **source program** is mcanalysis.pas, and this will then be compiled by the compiler into the **object program**, mcanalysis.exe.





## 4.4 Program Execution

Program file: mcanalysis.exe

Data file to be prepared for input (default):

- The text file storing the answer keys: ANSKEY.TXT
- The text file storing the participants' answers: MCDATA.TXT
- The text file storing the list of school: SCHOOL.TXT

All the program files and the text files aforementioned **should be put into the same folder** before execution if the user is trying to adopt the input method of using the keyboard. Or else, the program **cannot be executed successfully**.

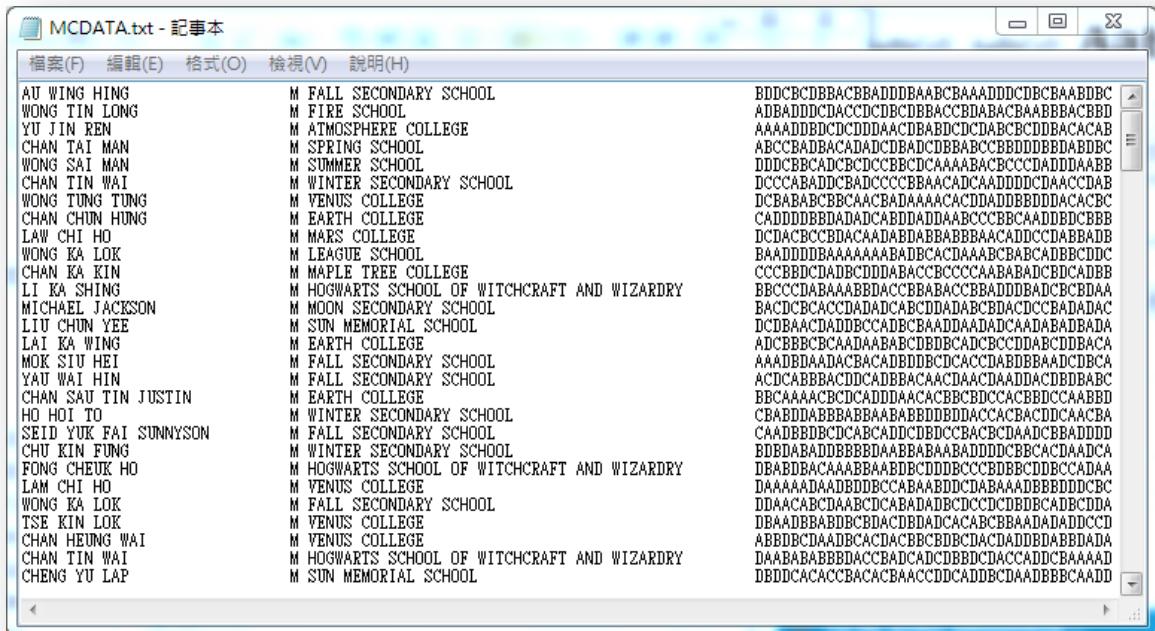
The text files are prepared by MS Notepad, below are the screenshots of the sample text files and some of the program interfaces:

To execute the program, the user should **prepare the text files** and then **open the program**.

**Instructions** and **prompt words** are provided by the program for telling the user what to do in the program.

## Multiple-Choice Analysis Report

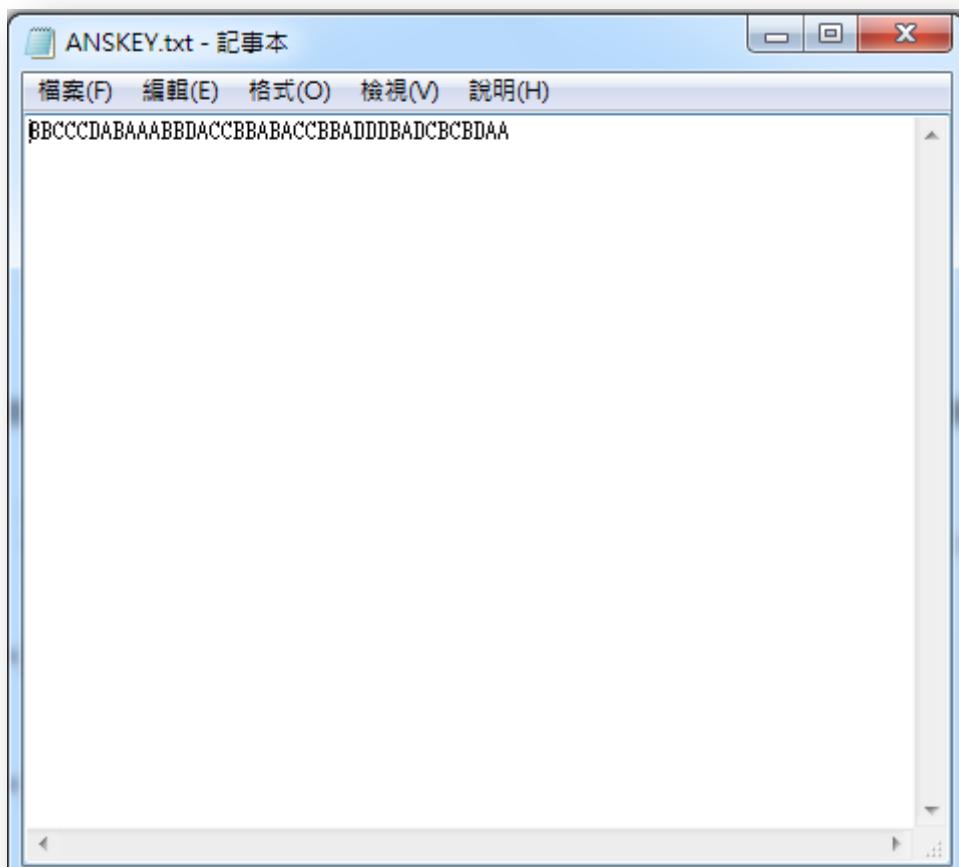
### 1. MCDATA.TXT



Participant Name	Answer	Text Content (Right Column)
AU WING HING	M FALL SECONDARY SCHOOL	BDDCBCBDBACBBADDDBAABCBAADDDBCBBAABDBCA ADBADDCCDACCDCDBCBDBACCBDBACBBAABBBACBBD
WONG TIN LONG	M FIRE SCHOOL	AAAADDDBCDPDDDAACDBABDCPDCABCBCDBACACAB
YU JIN REN	M ATMOSPHERE COLLEGE	ABCCBADBACADADCDBAIDCDBBACBCCBDBDEBDBDC
CHAN TAI MAN	M SPRING SCHOOL	DDDCBBCACDCBCDCBECDCIAAAABCBCCCDADDDAABB
WONG SAI MAN	M SUMMER SCHOOL	DCCCABADDCBACDCCCBBAAACDCAADDCCDAACCB
CHAN TIN WAI	M WINTER SECONDARY SCHOOL	DCBABAABCBCAACBAAAACACCDADDBBDDDACACBC
WONG TUNG TUNG	M VENUS COLLEGE	CADDDBBDBADADCABEDDAIDAABCCEBCAADDDBCBBB
CHAN CHUN HUNG	M EARTH COLLEGE	DCDACEBCBDCACAADABDABABBBAAACADDCCDABBBAD
LAW CHI HO	M MARS COLLEGE	BAADDDBDAAAAAAAPADBCACDAAAPCBABCACBDCDC
WONG KA LOK	M LEAGUE SCHOOL	CCCCBDCJADBCDJDABACCCBCCCCAABABADCBDCADBB
CHAN KA KIN	M MAPLE TREE COLLEGE	BBCCCDABAAABEDBACCBEBABACCBBAADDDBACBEBDAA
LI KA SHING	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	BACDCBCACCCDADACBDCDADABCBDACDCBCBACDADAC
MICHAEL JACKSON	M MOON SECONDARY SCHOOL	DCDBAACDADDBCCADDBCBAADDAAADCAADABADBADAA
LIU CHUN YEE	M SUN MEMORIAL SCHOOL	ADCBBCBCCAADAAAABABCDBDCACDBCCDDABCDBACA
LAI KA WING	M EARTH COLLEGE	AAADBDAAADACBACADEDDDBDCACCCDABDBBAACDCBCA
MOK SIU HEI	M FALL SECONDARY SCHOOL	ACDCABBBACDCDCAADBACACDAAACDADDBACBABC
YAU WAI HIN	M FALL SECONDARY SCHOOL	BBCCAAACACBCDCAADDAAACCEBCBDCCACBBDCAAABBD
CHAN SAU TIN JUSTIN	M EARTH COLLEGE	CBABDDAABBBABAABBBDDDACACBACDCACCAACBCA
HO HOI TO	M WINTER SECONDARY SCHOOL	CAADBBDCDCABCADDCCDIBDCBACBCDAAACDBBADDDB
SEID YUK FAI SUNNYSON	M FALL SECONDARY SCHOOL	BDBDABADDBBBBDAABBBABAABDDDCBBCACDAAADCA
CHU KIN FUNG	M WINTER SECONDARY SCHOOL	DBABDBACAAAABEAABDBCDDBBCCEDBBCDADCCADAA
FONG CHEUK HO	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	AAAAAAADADDDBCCABAABABDCDABAAABDBBDDDCBC
LAM CHI HO	M VENUS COLLEGE	DDAACABCDAABCDCABADABDCDCDCDBBDBACDBCDDA
WONG KA LOK	M FALL SECONDARY SCHOOL	DBAADDDBBABDBCBDAIDBDADDCACABCBBAAADADDCD
TSE KIN LOK	M VENUS COLLEGE	ABEDBCDAAADBACDACDBCCBCBDCDADDBBDBDADDA
CHAN HEUNG WAI	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	DAABABABBBDACCBAIDCDBDCACCCDADDBCDAAABBB
CHENG YU LAP	M SUN MEMORIAL SCHOOL	DBDDCACCCBACACBAAACCCDADDBCDADDBBCAADD

The participants' answers and information are inserted in a text file like this. The file name is not restricted but the format will exactly be the same as stated in Chapter 3

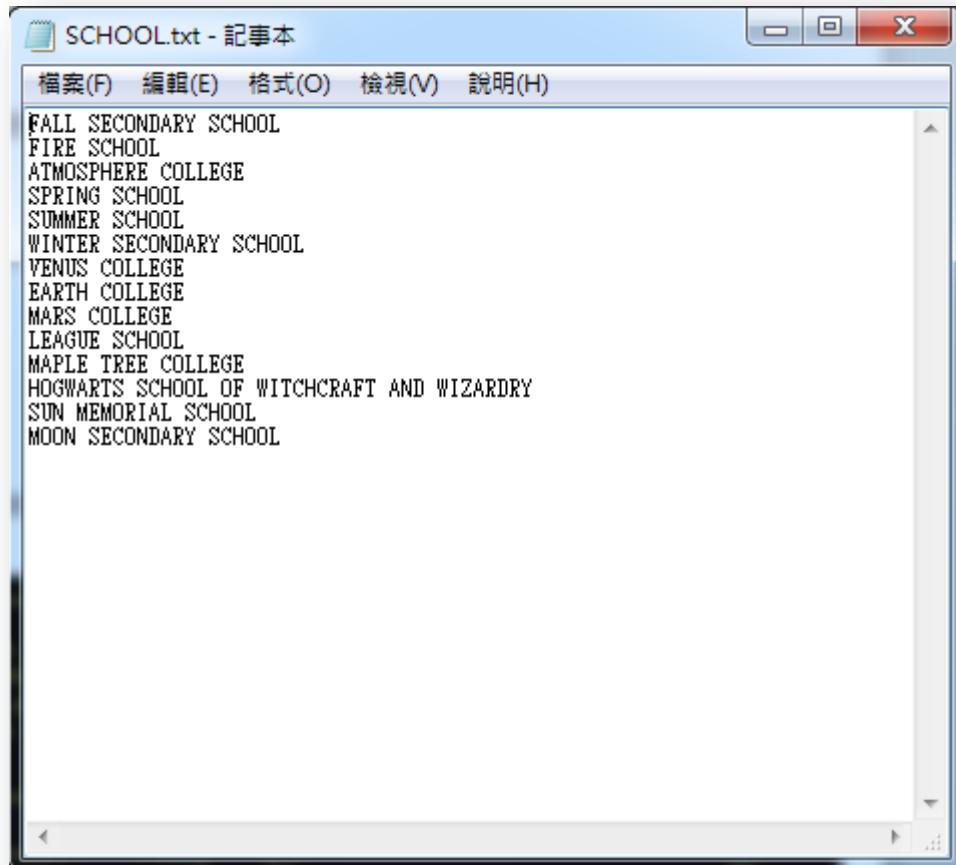
### 2. ANSKEY.TXT



## Multiple-Choice Analysis Report

The answer key is inserted in a text file like this. The file name is not restricted but the format will exactly be the same as discussed in Chapter 3.

### 3. SCHOOL.TXT



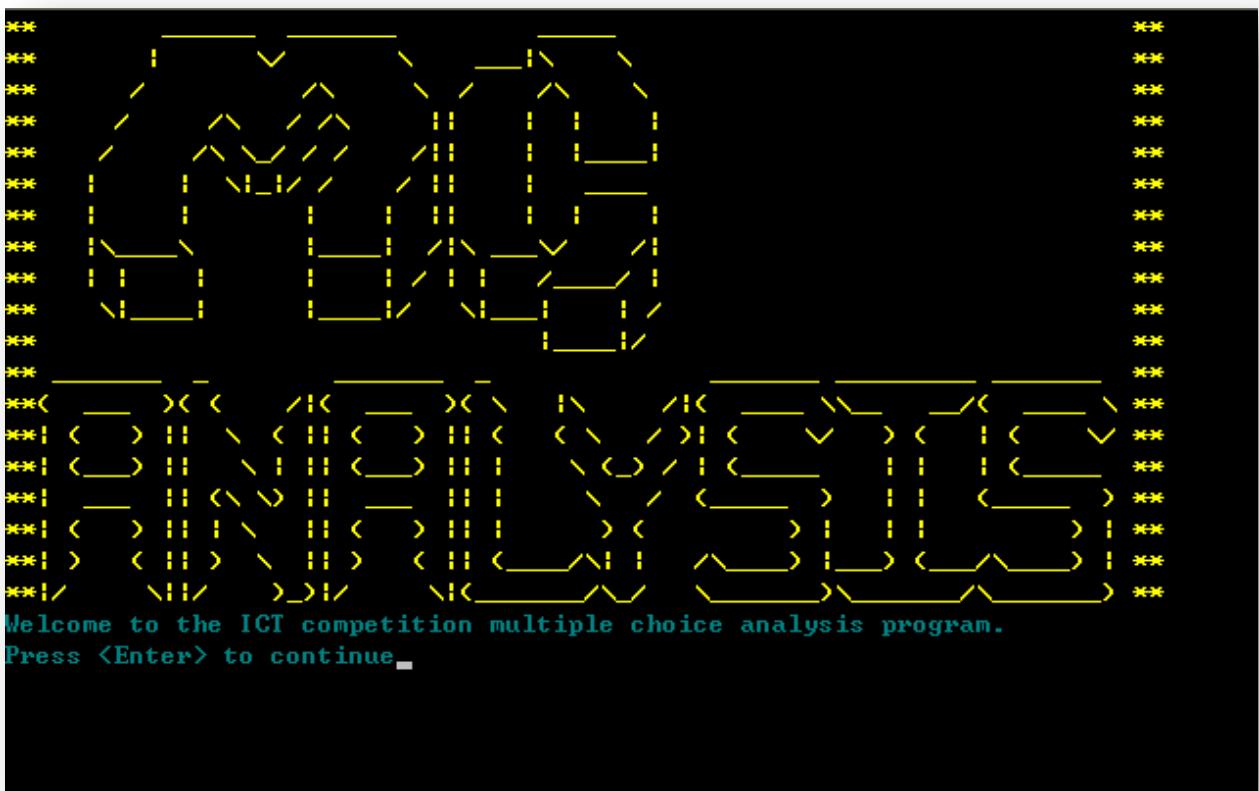
A screenshot of a Windows Notepad window titled "SCHOOL.txt - 記事本". The window contains a menu bar with "檔案(F)", "編輯(E)", "格式(O)", "檢視(V)", and "說明(H)". The main text area lists the following schools:

```
FALL SECONDARY SCHOOL
FIRE SCHOOL
ATMOSPHERE COLLEGE
SPRING SCHOOL
SUMMER SCHOOL
WINTER SECONDARY SCHOOL
VENUS COLLEGE
EARTH COLLEGE
MARS COLLEGE
LEAGUE SCHOOL
MAPLE TREE COLLEGE
HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY
SUN MEMORIAL SCHOOL
MOON SECONDARY SCHOOL
```

The list of school is inserted in a text file like this. The file name is not restricted but the format will exactly be the same as discussed in Chapter 3.



## 4. Program: Welcoming screen



The user will be welcomed on this screen.



## 5. Program: Inputting text files

```
8 8888 ,o888888o.8888888 8888888888
8 8888 8888 '88. 8 8888
8 8888 ,8 8888 '8. 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 '8 8888 .8 8 8888
8 8888 8888 ,88 8 8888
8 8888 '8888888P 8 8888      COMPETITION MC ANALYSIS
*****
Please input the file name of the participants' information and answers
default:<MC DATA.txt> : -
```

The user will input the text files needed for execution. They are the file of the participants' information and answers, the file of the answer key and the list of school. These files will be read into the program one by one and stored for further operations. The user can choose whether to input the text files by typing through keyboard or dragging the path in.



## 6. Program: Main menu user interface

The user will arrive at this main menu after inputting all the files. They can choose what to be displayed on screen simply by inputting a number from 1 to 13. Option 14 is for saving the report in text form while option 15 is for quitting the program.



## 7. Program: Displaying numbers of participants( e.g. total number of participants)

```
Numbers of participations
*****
1. Total number of participants
2. Total number of schools participated
3. Total number of participants from each schools

Questions and Score Analysis
*****
6. Percentage correct for each questions
7. Standard deviation
8. Mean/ Median /Mode
9. Range/ Upper,lower-quartile/ Inter-quartile range

Awardees
*****
4. Individual awardees
5. School awardees

Search
*****
10. By name
11. By school
12. By sex

13. Display the list of all participants
14. Save report in text form
15. Quit

Please enter the option<1-15>: 1

Total number of partipants for the competition: 200
Total number of male particpants: 118
Total number of female particpants: 82
Press <enter> to continue
```

Some results to be shown are short that they will be displayed under the menu, for example, the total number of participants is of the case. The information is displayed under the menu. The user can press <enter> to go back to the menu once again for another option.



## 8. Program: Displaying awardees (e.g. individual awardees)

Here is the list of individual awardees of the competition		
Name	Sex School	Award
LI KA SHING	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Medal
NG SIU YEE	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Medal
LEUNG HIU FUNG	M SPRING SCHOOL	Medal
HOWARD YOUNG	M FALL SECONDARY SCHOOL	Medal
SO PANG HIN	M MOON SECONDARY SCHOOL	Medal
ADAM HENDERSON	M UENUS COLLEGE	HD
CHAN TIN WAI	M WINTER SECONDARY SCHOOL	HD
FONG CHEUK HO	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	HD
CHAN PUI HAN	F MOON SECONDARY SCHOOL	HD
CYNTHIA MARTIN	F UENUS COLLEGE	HD
BRENDA RIVERA	F MARS COLLEGE	HD
JANICE BROOKS	F UENUS COLLEGE	HD
TAM WING YAN	F FALL SECONDARY SCHOOL	HD
CARL ALEXANDER	M MAPLE TREE COLLEGE	HD
TERESA JENKINS	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	HD
WONG NGA YIN BELINDA	F EARTH COLLEGE	HD
LEE SHU JEN	M LEAGUE SCHOOL	HD
LEUNG WAI CHI	M EARTH COLLEGE	HD
LUI KEI KWAN	M LEAGUE SCHOOL	HD
RAYMOND COX	M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	HD
MICHELLE CLARK	F SUN MEMORIAL SCHOOL	Credit
KAREN WILLIAMS	F SUN MEMORIAL SCHOOL	Credit
JULIA YOUNG	F MARS COLLEGE	Credit
KATHERINE MORGAN	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Credit
CHAN SIN LING CHARLOTTE	F MARS COLLEGE	Credit
LAU MING LEE CLARRISA	F WINTER SECONDARY SCHOOL	Credit
FUNG MO YEE	F SPRING SCHOOL	Credit
TAM WING SUM	F SPRING SCHOOL	Credit
NG CHI YANF	F LEAGUE SCHOOL	Credit
TSANG MAN KEI	F SPRING SCHOOL	Credit
LIU CHUN YEE	M SUN MEMORIAL SCHOOL	Credit
TSE KIN LOK	M UENUS COLLEGE	Credit
FUNG KWOK LUN	M MAPLE TREE COLLEGE	Credit
WONG KAM SHING	M WINTER SECONDARY SCHOOL	Credit
YIP WAI TING	M MARS COLLEGE	Credit
UICTOR COLLINS	M FALL SECONDARY SCHOOL	Credit
WALTER ROGERS	M MARS COLLEGE	Credit
BRANDON MARTINEZ	M WINTER SECONDARY SCHOOL	Credit
AARON COOK	M SUN MEMORIAL SCHOOL	Credit
ANNIE RODRIGUEZ	F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Credit
CAROLYN GREEN	F SUN MEMORIAL SCHOOL	Credit

Key: HD=High Distinction, D=Distinction

Press <enter> to continue

## Multiple-Choice Analysis Report

Some results are large in amount that they should be displayed on a full screen. For example, listing the individual awardees will need a full screen such that the user can read the list of awardees more easily as they are more tidily shown. The user can press <enter> to go back to the main menu.

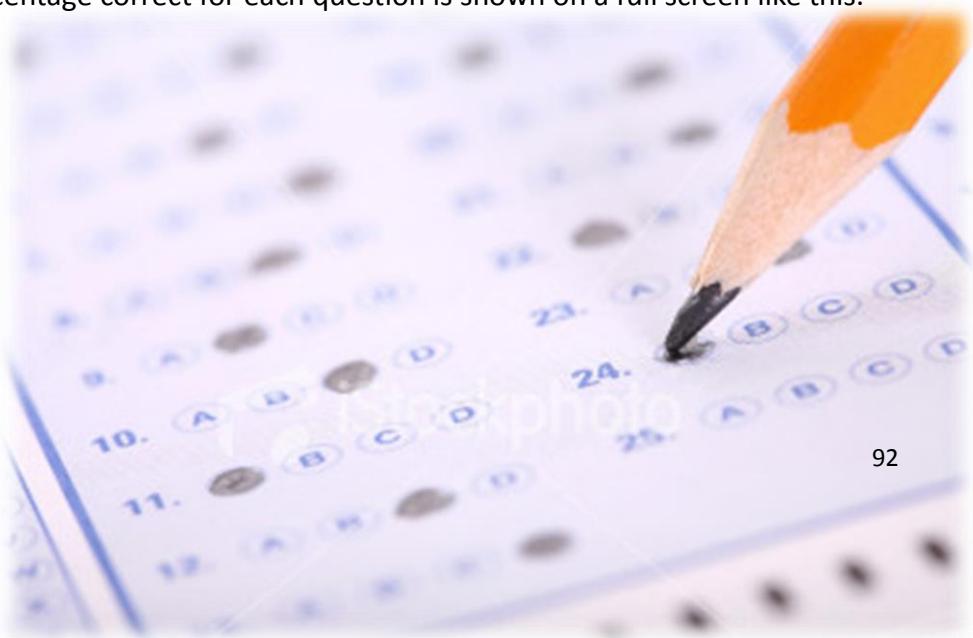
### 9. Program: Questions and score analysis (e.g. percentage

correct and standard deviation)

Question No.	% correct	Question No.	% correct
1	31.50	21	25.50
2	27.00	22	24.00
3	22.50	23	29.50
4	23.50	24	34.00
5	23.50	25	23.50
6	24.50	26	22.50
7	23.50	27	26.50
8	32.00	28	29.00
9	25.00	29	20.50
10	25.50	30	22.00
11	25.50	31	23.00
12	25.50	32	30.50
13	27.00	33	23.50
14	23.50	34	21.50
15	26.50	35	28.50
16	28.00	36	29.00
17	24.50	37	26.00
18	25.50	38	25.00
19	29.50	39	26.00
20	29.00	40	29.50

Press <enter> to continue

Some statistics like the percentage correct for each question is shown on a full screen like this.



## Multiple-Choice Analysis Report

```
!__\__! !__! COMPETITION MC ANALYSIS
*****
Numbers of participations          Awardees
*****                                *****
1. Total number of participants    4. Individual awardees
2. Total number of schools participated 5. School awardees
3. Total number of participants from each schools

Questions and Score Analysis        Search
*****                                *****
6. Percentage correct for each questions 10. By name
7. Standard deviation               11. By school
8. Mean/ Median /Mode              12. By sex
9. Range/ Upper-lower-quartile/ Inter-quartile range

13. Display the list of all participants
14. Save report in text form
15. Quit
Please enter the option(1-15): 7
Standard deviation of score is: 29.57
Press <enter> to continue
```

And some of the short piece of statistics, e.g. standard deviation of score, is shown like this under the menu.



#### 10. Program: Seacrching ( e.g. by name)

|\_\_\\_\_|\_|\_| COMPETITION MC ANALYSIS

\*\*\*\*\*

**Numbers of participations**

\*\*\*\*\*

1. Total number of participants

2. Total number of schools participated

3. Total number of participants from each schools

**Awardees**

\*\*\*\*\*

4. Individual awardees

5. School awardees

**Questions and Score Analysis**

\*\*\*\*\*

6. Percentage correct for each questions

7. Standard deviation

8. Mean/ Median /Mode

9. Range/ Upper,lower-quartile/ Inter-quartile range

**Search**

\*\*\*\*\*

10. By name

11. By school

12. By sex

13. Display the list of all participants

14. Save report in text form

15. Quit

Please enter the option(1-15): 10

Please enter the name of participant:  
LI YO SHING

Here is the record you want

\*\*\*\*\*

Name	Sex	School	Score
*****	*****	*****	*****
LI KA SHING	M	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	40

Press <enter> to continue

The user can search for a particular participant by part of their information. For example, they can search for it by their names. The screenshots above show how it is done

## 11.Program: Displaying the list of all participants

Name	Sex	School	Score
*****			
LI KA SHING	M	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	40
NG SIU YEE	F	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	35
LEUNG HIU FUNG	M	SPRING SCHOOL	32
HOWARD YOUNG	M	FALL SECONDARY SCHOOL	18
SO PANG HIN	M	MOON SECONDARY SCHOOL	17
ADAM HENDERSON	M	VENUS COLLEGE	17
CHAN IIN WAI	M	WINTER SECONDARY SCHOOL	16
PONG CHEUK HO	M	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	16
CHAN PUI HAN	F	MOON SECONDARY SCHOOL	16
CYNTHIA MARTIN	F	VENUS COLLEGE	16
BRENDA RIVERA	F	MARS COLLEGE	16
JANICE BROOKS	F	VENUS COLLEGE	16
TAM WING YAN	F	FALL SECONDARY SCHOOL	16
CARL ALEXANDER	M	MAPLE TREE COLLEGE	15
TERESA JENKINS	F	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	15
WONG NGA YIN BELINDA	F	EARTH COLLEGE	15
LEE SHU JEN	M	LEAGUE SCHOOL	14
LEUNG WAI CHI	M	EARTH COLLEGE	14
LUI KEI KWAN	M	LEAGUE SCHOOL	14
RAYMOND COX	M	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	14

The list of all participants can be shown by inputting 13 for the option. Then the participants will be shown with descending scores.



## 12.Program: Saving report in text form

```
          + + + + +  
          | | | | | COMPETITION MC ANALYSIS  
*****  
Numbers of participations  
*****  
1. Total number of participants  
2. Total number of schools participated  
3. Total number of participants from each schools  
  
Questions and Score Analysis  
*****  
6. Percentage correct for each questions  
7. Standard deviation  
8. Mean/ Median /Mode  
9. Range/ Upper,lower-quartile/ Inter-quartile range  
  
13. Display the list of all participants  
14. Save report in text form  
15. Quit  
Please enter the option<1-15>: 14
```

Enter the name of the text file you want to save as (with .txt):

```
          + + + + +  
          | | | | | COMPETITION MC ANALYSIS  
*****  
Numbers of participations  
*****  
1. Total number of participants  
2. Total number of schools participated  
3. Total number of participants from each schools  
  
Questions and Score Analysis  
*****  
6. Percentage correct for each questions  
7. Standard deviation  
8. Mean/ Median /Mode  
9. Range/ Upper,lower-quartile/ Inter-quartile range  
  
13. Display the list of all participants  
14. Save report in text form  
15. Quit  
Please enter the option<1-15>: 14
```

Enter the name of the text file you want to save as (with .txt): report.txt  
Loading...Success

Press <enter> to continue

The user may save all the analysis results in text form by inputting 14 for the option.

## 13. Output text file of the report

report.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

A Report on the ICT competition

Participation:

\*\*\*\*\*

Total number of participants for the competition: 200  
Total number of male participants: 118  
Total number of female participants: 82  
Total number of schools for the competition: 14

Total number of participants from each schools

\*\*\*\*\*

School	No. of participants
SUMMER SCHOOL	1
FIRE SCHOOL	1
ATMOSPHERE COLLEGE	1
LEAGUE SCHOOL	15
MOON SECONDARY SCHOOL	14
EARTH COLLEGE	15
MAPLE TREE COLLEGE	15
SPRING SCHOOL	15
MARS COLLEGE	18
FALL SECONDARY SCHOOL	16
VENUS COLLEGE	19
SUN MEMORIAL SCHOOL	23
WINTER SECONDARY SCHOOL	22
HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	25

AWARDS:

Individual awardees of the competition

\*\*\*\*\*

Name	Sex	School	Award
LI KA SHING	M	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Medal
NG SIU YEE	F	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	Medal
LEUNG HIU FUNG	M	SPRING SCHOOL	Medal
HOWARD YOUNG	M	FALL SECONDARY SCHOOL	Medal
SO PANG HIN	M	MOON SECONDARY SCHOOL	Medal
ADAM HENDERSON	M	VENUS COLLEGE	HD
CHAN TIN WAI	M	WINTER SECONDARY SCHOOL	HD
FONG CHEUK HO	M	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	HD
CHAN PUI HAN	F	MOON SECONDARY SCHOOL	HD
CYNTHIA MARTIN	F	VENUS COLLEGE	HD
BRENDA RIVERA	F	MARS COLLEGE	HD
JANICE BROOKS	F	VENUS COLLEGE	HD
TAM WING YAN	F	FALL SECONDARY SCHOOL	HD
CARL ALEXANDER	M	MAPLE TREE COLLEGE	HD
TERESA JENKINS	F	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	HD
WONG NGA YIN BELINDA	F	EARTH COLLEGE	HD
LEE SHU JEN	M	LEAGUE SCHOOL	HD
LEUNG WAI CHI	M	EARTH COLLEGE	HD
LUI KEI KWAN	M	LEAGUE SCHOOL	HD
RAYMOND COX	M	HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY	HD
SENA KOBAYAKAWA	M	MAPLE TREE COLLEGE	HD
MARYLIN RUSSELL	F	VENUS COLLEGE	HD
KATHY BAKER	F	SUN MEMORIAL SCHOOL	HD
CHAN WING YEE	F	SPRING SCHOOL	HD
AU WING HING	M	FALL SECONDARY SCHOOL	HD
CHU KIN FUNG	M	WINTER SECONDARY SCHOOL	HD

# **Chapter 5 Testing & Evaluation**

## **5.1 Brief Description**

It is important to test whether the program can **run smoothly in different conditions**, especially as the program is designed for the user to use in nature, testing and evaluation by both the programmer and the users are essential.

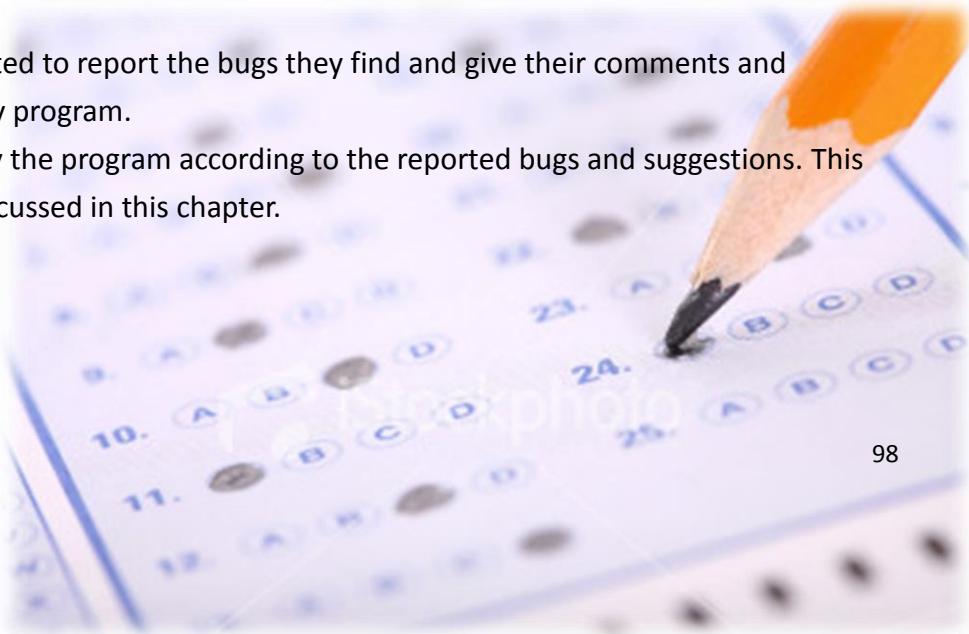
In this chapter, the followings will be done:

- to find out the bugs (syntax, logical and run-time errors) in the program,
- to check whether the program can achieve its purposes,
- to see whether the program is user-friendly, compatible and flexible
- to debug and improve the program based on the testing and evaluation results

## **5.2 Testing and Evaluation Plan**

The program will be tested and evaluated according to the following plan:

- ⌚ Internal testing and evaluation
  - ⌚ The program will be tested intensively by the programmer
  - ⌚ Different test cases are devised to test the program thoroughly
  - ⌚ Tests using correct data, incorrect data and extreme data are conducted.
  - ⌚ The program will also be evaluated according to its user-friendliness, compatibility, flexibility for future development, reusability of program codes, etc.
- ⌚ External testing and evaluation:
  - ⌚ 10 external users, i.e. friends in my school will be invited to test and evaluate the program.
  - ⌚ The object program will be sent to them through e-mail or some communication software.
  - ⌚ The users are invited to report the bugs they find and give their comments and suggestions on my program.
  - ⌚ I will try to modify the program according to the reported bugs and suggestions. This will be further discussed in this chapter.



## 5.3 Internal testing

Internal testing is mainly for testing out **syntax, logical** and **run-time errors** and testing whether the program can achieve its purpose.

**Syntax errors** can be automatically detected by the **program compiler** whereas the **other two kinds of errors** need programmer's effort to be tested by inputting different **valid** and **invalid** data.

### 1. Testing for syntax error

The program compiler, Dev-pascal will check automatically for syntax error, for example,

```
9 / 5 mcanal~1.pas  
Fatal: Syntax error, ; expected but identifier PSCORE found
```

when some kind of syntax error like this is mentioned by the compiler, the error can be **easily fixed** by checking the program line where the error occurred at. In the above case, a semi-colon was left, so the problem is fixed by typing a ";" in.

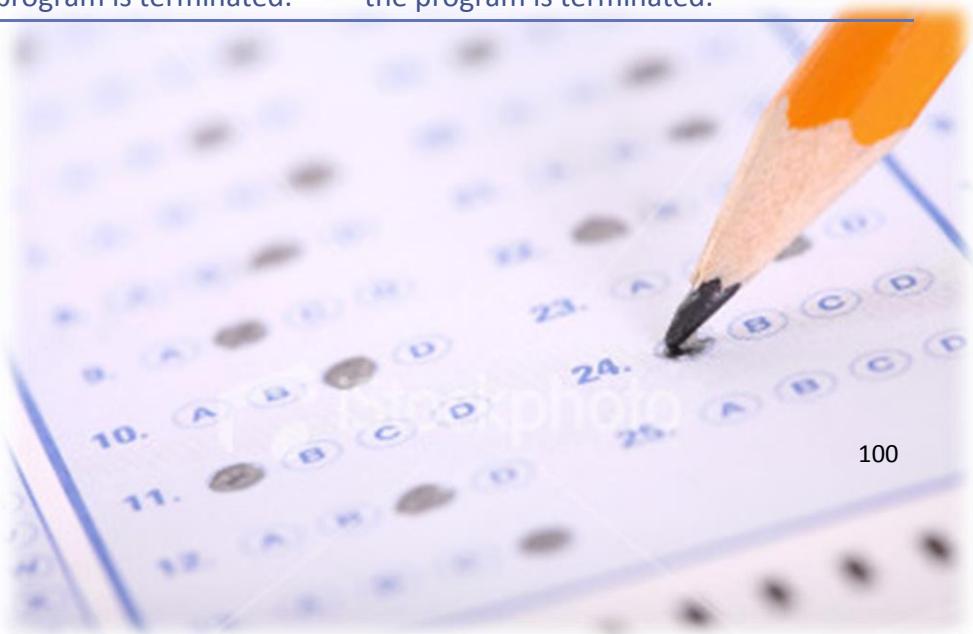


## 2. Testing for the procedures in the program

### 1. Correct data input

First of all, **correct input** of all the data provided by the user described in the previous chapters is tested.

Testing procedures	Expected Outcome	Actual Outcome	Remarks
<b>Welcoming screen</b>	The welcoming screen is displayed as normal	The welcoming screen is displayed as normal	PASS
<b>Inputting text files and sorting</b>	The sorted data is successfully read and can be shown on the screen	The sorted data is successfully read and can be shown on the screen	PASS
<b>Main menu</b>	The user can choose an option and go back to the main menu by pressing enter	The user can choose an option and go back to the main menu by pressing enter	PASS
<b>Showing on-screen results</b>	All of the statistical results are correct	All of the statistical results are correct( including total number of participants, schools and % correct for each question, etc.)	PASS
<b>Exporting text file</b>	All of the calculated statistical results correspond to the on-screen results	All of the calculated statistical results correspond to the on-screen results	PASS
<b>Quit</b>	The ending animation is successfully shown and then the program is terminated.	The ending animation is successfully shown and then the program is terminated.	PASS



## 2. Invalid data input

Secondly, some invalid data are inputted in case the user has inputted **wrong data** into the program.

### Test case 1

Purpose: To check whether there is error when inputting file names

Input:

- The file names inputted are of integer type/ file name that does not exist in the same folder
- An empty string is inputted
- Files other than .txt is inputted
- Accidentally, a file path with “” at two ends has been read when the file path was dragged in.

Expected The program can detect the error and tell the user to input again

Output:

Actual

- If an empty string is inputted at first, the program continues to be executed but a new input cannot be read
- If files other than .txt are inputted, the program will still accept it but produce “mojibake”, which means random characters.
- “” at two ends make file name reading unsuccessful.

Output: FAIL, bugs found

Result:

Follow-up Modification is done to fix the inputting mechanism.

Action:

```
8 8888      ,o888888o.88888888 8888888888
8 8888  8888     '88.    8 8888
8 8888 ,8 8888     '8.    8 8888
8 8888 88 8888           8 8888
8 8888 88 8888           8 8888
8 8888 88 8888           8 8888
8 8888 88 8888           8 8888
8 8888 '8 8888       .8   8 8888
8 8888     8888     ,88   8 8888
8 8888     '8888888P   8 8888      COMPETITION MC ANALYSIS
*****
Please input the file name of the participants' information and answers
default:<MCMDATA.txt> :HAHAHA
The file required to be opened is not found.
```

# Multiple-Choice Analysis Report

```
8 8888 ,o888888o.88888888 8888888888
8 8888 8888 '8. 8 8888
8 8888 ,8 8888 '8. 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 '8 8888 .8 8 8888
8 8888 8888 ,88 8 8888
8 8888 '8888888P 8 8888      COMPETITION MC ANALYSIS
*****
Please input the file name of the participants' information and answers
default:<MCDATA.txt> :HAHAHA
The file required to be opened is not found.

Please input the file name of the participants' information and answers
default:<MCDATA.txt> :

MCData.txt
MCData.txt
```

```
8 8888 ,o888888o.8888888 8888888888
8 8888 8888 '88. 8 8888
8 8888 ,8 8888 '8. 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 8 8888
8 8888 88 8888 .8 8 8888
8 8888 8888 ,88 8 8888
8 8888 '8888888P 8 8888      COMPETITION MC ANALYSIS
*****
Please input the file name of the participants' information and answers
default:<MC DATA.txt> :C:\Users\admin\Desktop\Untitled.jpg
Please input the file name of the answer key
default:<ANS KEY.txt> :ANSKEY.TXT
Please input the file name of school list
default:<SCHOOL.txt> :SCHOOL.TXT
```

## Multiple-Choice Analysis Report

```

H W f~ g~ K Y / + ↓ vkkH H? e↓ - ↓ ↓
GGG ± Z 6↓e F-t #↓ < - -4 Jh / 1<x^= Wx<w△*?'
Z < >↓@ *4Mm*L△ *↑p⊕+@ HbK ♦- & 0c H△, [↓ 0 vdc ? - # = N
:q↓* ± *4*4+ -1' Z↓< K → * E↑ ↓ Gy < c Y] - L ↓ .
:q↓* ± *4*4+ -1' Z↓ ↓ R @S-L↓ Y] " < 1~↓ > R=9 *ze 4Bt L J
^*+ + qL↓ I;I 5, , ↓, 1+ +↑ m/ △ !↓ j z ↓△ R
△+ < 6 i >‡ ^ 6 y 9 0 r L 3 f +flk ↓ +2 d lg δ M↓ >
2 \ > *c >? 8 z4 F > p=1↓ 2 ↓ $ zh _Fq +i- \ ↓-
$0 1 4Y 3E ZM# y < : 0041 K G ↓ 6 x ^ :qL↓ & q -0
Qan * u b ↓ L < 4n8 xs↓4x Q ? 6 ↓ P -k<- xs↓ 3 qo <
S ↓ 1 [ $↓ 5 *D > cS #-Ik‡ $ W 9M "1 F J + v<
↓c +P W 4 1 43 !! Sk↓ 5 ♦ xg < 4ka B:1 -/ rW> ♦ ↓
↓ >? *- N y↓ Tq ↓ S tt8 ' ↓ ↓ s < o +c 2 ' 0 0 5*↓
q !L +↓ < -t w$- D Q =C ↓ ↓ +]*= -↓:
    -xq,?> - "u v ← >=
i <i< ↓ ! x<? W ↓ [ 2 ↓ m p ↓ -g↓ g ← C W ^@ & -c
N ↓ v↓ ↑ ▲ C > 0 *x 9I > +g ± ↑ m p 4<m5
3 < c 2 e↓ - 4nG 93 5 F <+* - K ! _↓ *q Q 5
↓ c h < e↓ - v- < u↓ /qm*↓ > ↓ vkkH H?
u M! ~@ & ^ !! 0 C-14!↓ ↓ c+ ↓△ Y* < vkkH H?
+ h=_ x Q ↓ 4G C ↓ Z S !+ I 48 L- LZ E-↓ ↓' !E43z -!! 0 C-1↓
[] ↓ c+ ↓△ Y* < vkkH H? -e↓ - v- < u↓ /qm*↓ > ↓ vkkH H?
-e↓ - 4nG 93 5 F <+* - K ! _↓ *q Q- > 0 *x 9I > +g
& * ↓ ↓ ♦ H " L ↓c- 04 48 f_uL E↑h -8 0 ↓ E <+\
< h ' K 8 ↓gh ?_ ↓ _ ↓ Lr x -M9↓△ Js H↓
fG N I W ~ 4 + 5 ↓ 4G L T 4S 3s↓- ↓ - L ↓ * e * /
? -Z△ H ↓ # A △ W_‡
△ 1J ↓ *△~ o 0 ? C\↓ g Ro-↓↑ *o → m*T > ;4g Ro-↓ 0 Ix
-x ↓ 4+↓+ >1 ↓ Q C- 1U↓ < PrG-1Cj <↑
-x ↓ 4+4GQ @↓ Q @ s e * Ba [↓↓ ↓ m
-x ↓ @ Q @↓ Q @ 1 X v-U E 42W TG C - < #L- w>
f 4J5< J ^6 g-! A @ y 7< q - @ j< ↓ ↓ - G ♦ L'-M e\5 o [↓ JP4<
-* ZZ= 0. 0↓ #; X v < @>[ h v bm#X2; @UKnBtm △ -Q v
↓ 7sjg ↓↓↓↓ v △ U _G u p ↓↓↓↓↓↓! - < @>[ h v bm#X2; @UKnB
m △ -Q v-?< q - @ j< ↓ ↓ - G ♦ L'-M e\5 o [↓ JP↓
Data successfully read
Press <Enter> to enter the main menu

```

The third screenshot shows the error detected.



## Multiple-Choice Analysis Report

### Test case 2

**Purpose:** To check whether there is error when inputting options in the main menu

**Input:** The file names inputted are of integer type <1 or >15, characters and strings are also inputted

**Expected** The program can identify wrong input and tell the user to correct it.

**Output:**

**Actual** The program can identify wrong input and tell the user to correct it.

**Output:**

**Test** PASS

**Result:**

**Follow-up** NIL

**Action:**

```
Questions and Score Analysis
*****
6. Percentage correct for each questions
7. Standard deviation
8. Mean/ Median /Mode
9. Range/ Upper,lower-quartile/ Inter-quartile range
Search
*****
10. By name
11. By school
12. By sex

13. Display the list of all participants
14. Save report in text form
15. Quit
Please enter the option<1-15>: 0
Invalid input! Please try again.
Please enter the option<1-15>: 16
Invalid input! Please try again.
Please enter the option<1-15>: hahaah
Invalid input! Please try again.
Please enter the option<1-15>: asd
Invalid input! Please try again.
Please enter the option<1-15>: a
Invalid input! Please try again.
Please enter the option<1-15>: \
Invalid input! Please try again.
Please enter the option<1-15>:
```



## Multiple-Choice Analysis Report

### Test case 3

**Purpose:** To check whether there is error when calculating the score of each participant

**Input:** Answers other than 'A','B','C','D' are included in the text file containing participants' answers

**Expected** The program only regard 'A','B','C','D' as the scoring answers,

**Output:** neglecting other invalid answers

**Actual** The program only regard 'A','B','C','D' as the scoring answers,

**Output:** neglecting other invalid answers

**Test** PASS

**Result:**

**Follow-up** NIL

**Action:**



## Multiple-Choice Analysis Report

### Test case 4

**Purpose:** To check whether there is error when inputting name to search for a particular participant by name

**Input:**

- Names other than the existing participant
- Integers

**Expected Output:** The program identifies no record matches the target and returns no record found

**Actual Output:** The program identifies no record matches the target and returns no record found

**Test Result:** PASS

**Result:**

**Follow-up Action:** NIL

Action:

```
1_/_\_|_|_| COMPETITION MC ANALYSIS
*****
Numbers of participations
*****
1. Total number of participants
2. Total number of schools participated
3. Total number of participants from each schools
Awardees
*****
4. Individual awardees
5. School awardees
Questions and Score Analysis
*****
6. Percentage correct for each questions
7. Standard deviation
8. Mean/ Median /Mode
9. Range/ Upper,lower-quartile/ Inter-quartile range
Search
*****
10. By name
11. By school
12. By sex
13. Display the list of all participants
14. Save report in text form
15. Quit
Please enter the option(1-15): 10

Please enter the name of participant:
LEUNG SIU MING
```

```
Here is the record you want
*****
Name          Sex  School           Score
*****
RECORD NOT FOUND
Press <enter> to continue
```

## Multiple-Choice Analysis Report

### Test case 5

**Purpose:** To check whether there is error when inputting school name to search for a particular participant by school

**Input:**

- Names other than the existing schools
- Integers

**Expected Output:** The program identifies no record matches the target and returns no record found

**Actual Output:** The program identifies no record matches the target and returns no record found

**Test** PASS

**Result:**

**Follow-up** NIL

**Action:**

```
1_ _\_\_! 1_1 COMPETITION MC ANALYSIS
*****
Numbers of participations          Awardees
*****                                *****
1. Total number of participants    4. Individual awardees
2. Total number of schools participated 5. School awardees
3. Total number of participants from each schools

Questions and Score Analysis        Search
*****                                *****
6. Percentage correct for each questions 10. By name
7. Standard deviation             11. By school
8. Mean/ Median /Mode            12. By sex
9. Range/ Upper,lower-quartile/ Inter-quartile range

13. Display the list of all participants
14. Save report in text form
15. Quit
Please enter the option<1-15>: 11

Please enter the school of participant:
CHEUNG SHA WAN CATHOLIC SECONDARY SCHOOL
```

```
Here is the record you want
*****
Name          Sex  School           Score
*****          *****          *****
RECORD NOT FOUND

Press <enter> to continue
```

## Multiple-Choice Analysis Report

### Test case 6

**Purpose:** To check whether there is error when inputting sex to search for a particular participant by sex

**Input:**

- Characters other than 'M' , 'F', 'm' and 'f'
- Integer

**Expected** The program identifies the error and tell the user to correct it

**Output:**

**Actual** The program identifies the error and tell the user to correct it

**Output:**

**Test** PASS

**Result:**

**Follow-up** NIL

**Action:**

```
13. Display the list of all participants
14. Save report in text form
15. Quit
Please enter the option<1-15>: 12

Please enter the sex of participant<M/F>:
1
Invalid input, please enter M/F only
Please enter the sex of participant<M/F>:
-13
Invalid input, please enter M/F only
Please enter the sex of participant<M/F>:
HAHAHAHAA
Invalid input, please enter M/F only
Please enter the sex of participant<M/F>:
A
Invalid input, please enter M/F only
Please enter the sex of participant<M/F>:
\
Invalid input, please enter M/F only
Please enter the sex of participant<M/F>:
```



### 3. Extreme data input

Different sets of **extreme data input** are used to test whether the program can handle extreme data well, although they are not invalid input, sometimes a program cannot handle them well.

The testing section will be focused in the **text file** of participants' answers and information  
The testing results are summarized as the table below:

Description	Outcome	Remarks
<b>1 participant</b>	<ul style="list-style-type: none"> <li>- Schools got 0 score can also be awarded as 2<sup>nd</sup> runner up</li> <li>- Other functions work properly with information corresponds to the text file</li> </ul>	<b>FAIL</b> , though this extreme data can be omitted as analyzing one datum is meaningless
<b>200 participants</b>	All participants' information , individual and school awardees, statistical information are normally displayed	PASS
<b>All participants have the correct answers</b>	<ul style="list-style-type: none"> <li>- Not all of the participants have the same medal award</li> <li>- Other functions work properly with information corresponds to the text file</li> </ul>	<b>FAIL</b> <ul style="list-style-type: none"> <li>- though this extreme data can be omitted as this is nearly impossible in real situation</li> <li>- and it is out of my ability to correct it</li> </ul>
<b>All participants got wrong answers</b>	<ul style="list-style-type: none"> <li>- Not all of the participants have the same medal award</li> <li>- Other functions work properly with information corresponds to the text file</li> </ul>	<b>FAIL</b> , <ul style="list-style-type: none"> <li>- though this extreme data can be omitted as this is nearly impossible in real situation</li> <li>- and it is out of my ability to correct it</li> </ul>
<b>All participants are male</b>	All participants' information , individual and school awardees, statistical information are normally displayed	PASS

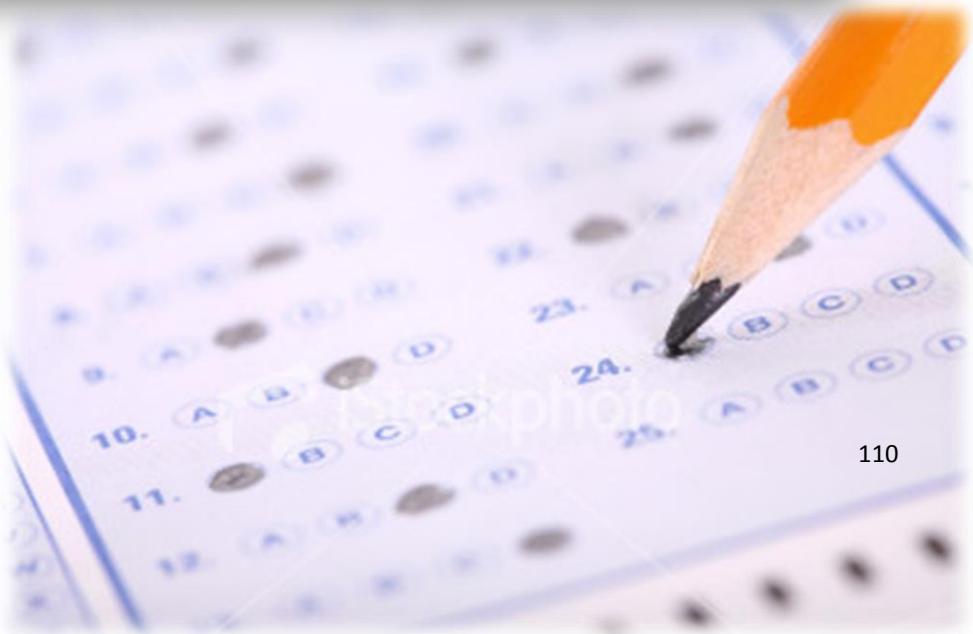
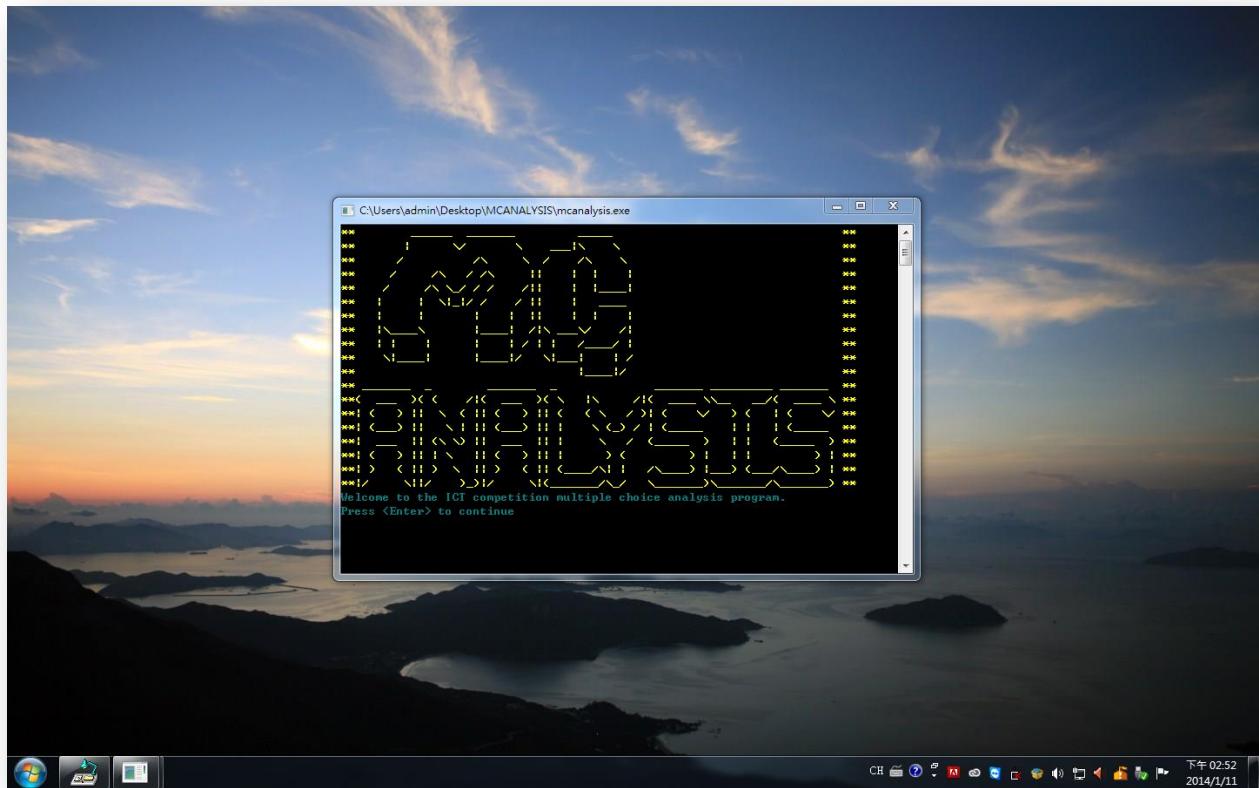
## Multiple-Choice Analysis Report

All participants are female	All participants' information , individual and school awardees, statistical information are normally displayed	PASS
-----------------------------	--	------

## 5.4 Compatibility testing

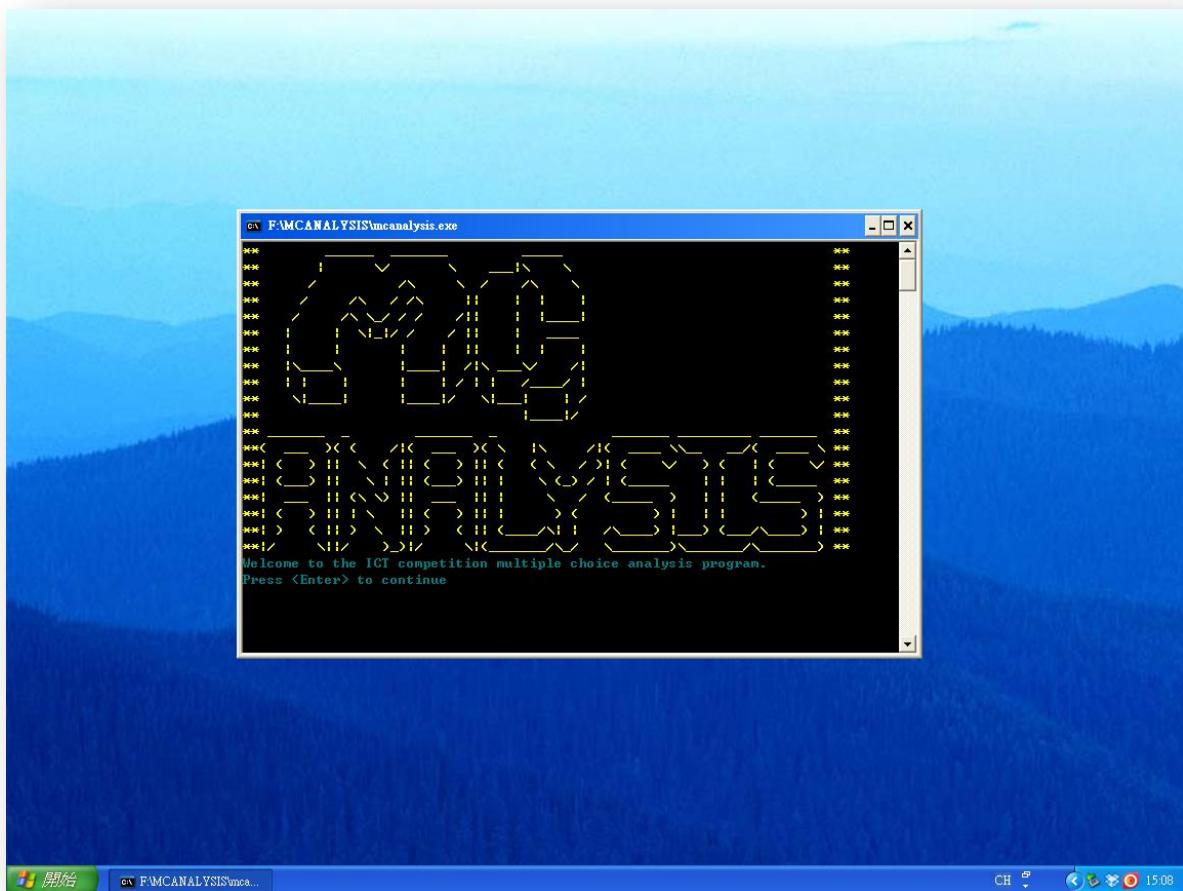
As mentioned in Chapter 2, this program is designed mainly for the usage on **Window 7**, **Window XP** and **Window 8**. After testing, the **object program** runs well on the three platforms, here are the screenshots:

Window 7:

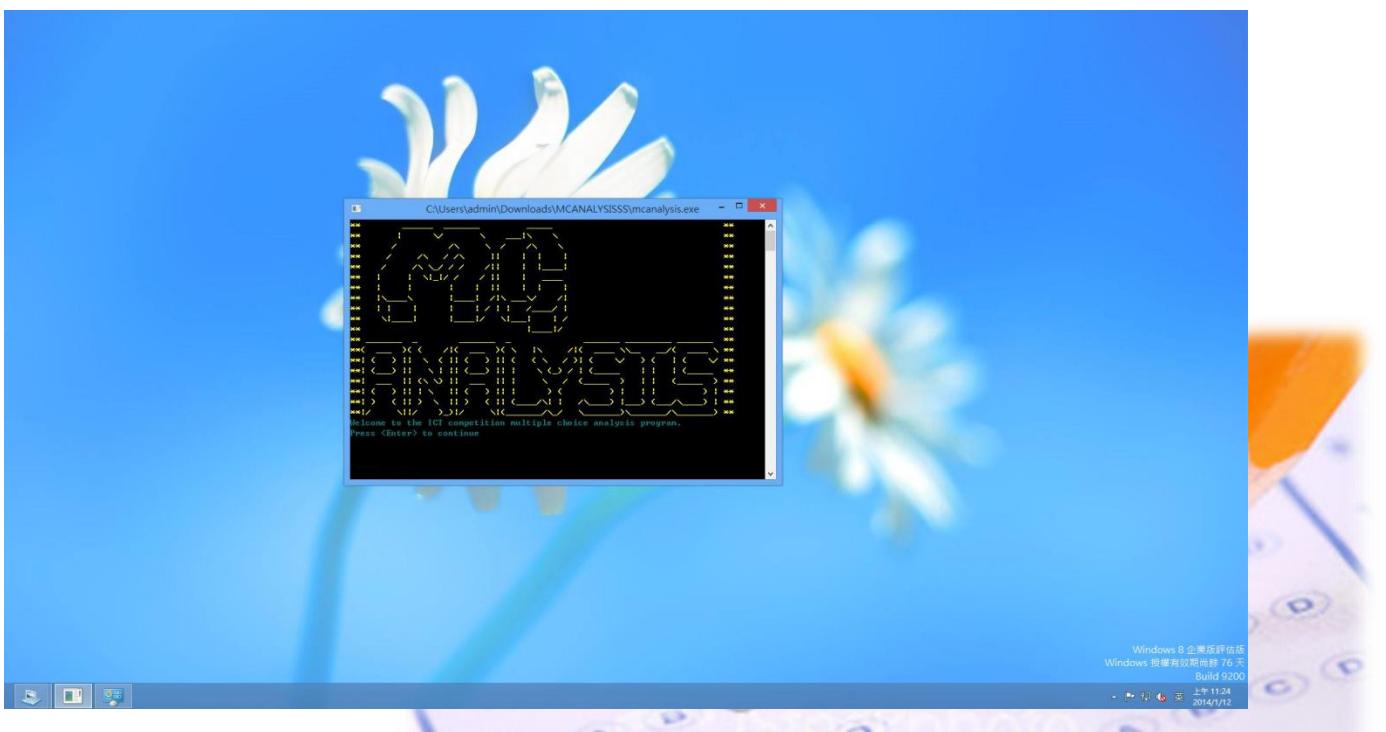


# Multiple-Choice Analysis Report

Window XP:



Window 8:



## 5.5 Debugging after testing

From the above testing results, it can be concluded that only **test case 1** in testing using **Invalid data input**, which is, when the program is reading the file name, If an **empty string** is inputted at first, the program **continues to be executed** but a new input cannot be read, and if a **non-text file is inputted**, the program **still accepts it and produces useless results**. Also, if the string inputted **contains "" at two ends**, the file name **cannot be read correctly**. So we need **debugging** for this.

After checking, the source of error is that, the mechanism of checking with IOResult cannot detect the **error of empty string**, so I have added an extra **REPEAT loop** in the original one to solve this.

```

repeat
    textColor(3);
    writeln('Please input the file name of the participants'''
information and answers');
    write('default:(MCDATA.txt) :');
    textColor(15);           {reading the participants' info}
    flag:=false;
    readln(pinfo);
    if (copy(pinfo,1,1)='') and (copy(pinfo,length(pinfo),1)='')
//removing ""
    then pinfo:=copy(pinfo,2,length(pinfo)-2);
    if NOT(copy(pinfo,length(pinfo)-2,3)='txt') and
NOT(copy(pinfo,length(pinfo)-2,3)='TXT') //checking extension
    then begin
        textColor(3);
        writeln('The file required is not a TXT file');

    end
else flag:=true;

until flag;
flag:=false;
```

In that sense, **"" at two ends will be immediately removed** after being read. And **empty string will not be accepted** by the program and so the user will be forced to input a correct one. Also, **only extension of txt is accepted**. The flag is reassigned to false for checking whether the file

## Multiple-Choice Analysis Report

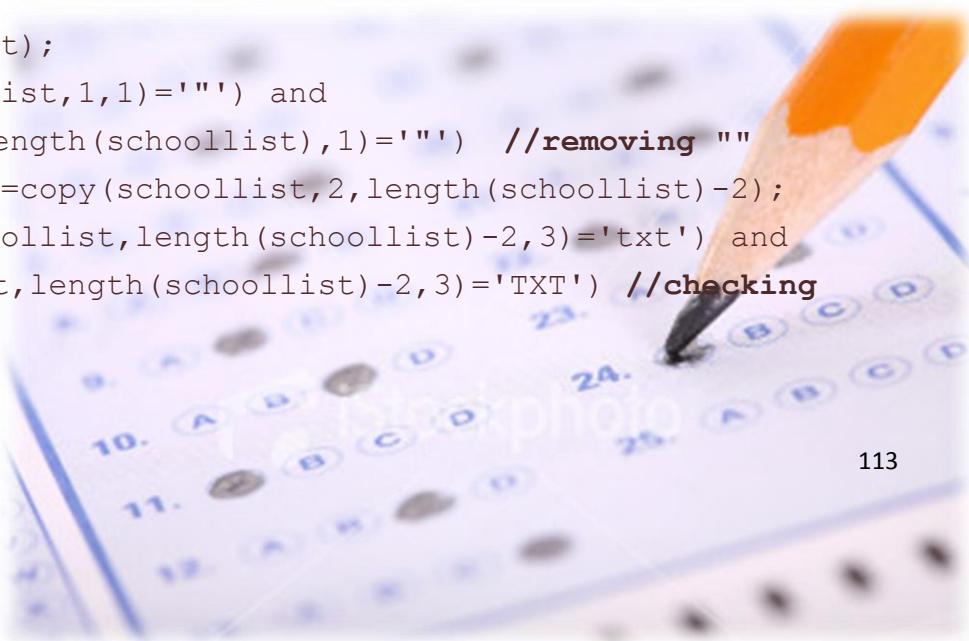
exists in the next section of the program.

The same is done to the input of answer key and the school list:

```
repeat
    textColor(3);
    writeln('Please input the file name of the answer key');
    write('default:(ANSKEY.txt) :');
    textColor(15);           {reading the anskey}
    flag:=false;
    readln(ans);
    if (copy(ans,1,1)='') and (copy(ans,length(ans),1)=='')
//removing ""
    then ans:=copy(ans,2,length(ans)-2);
    if NOT(copy(ans,length(pinfo)-2,3)='txt') and
NOT(copy(ans,length(ans)-2,3)='TXT')//checking extension
    then begin
        textColor(3);
        writeln('The file required is not a TXT file');
        end
    else flag:=true;

    until flag;
    flag:=false;

repeat
    textColor(3);
    writeln('Please input the file name of school list');
    write('default:(SCHOOL.txt) :');
    textColor(15);           {reading the participants' info}
    flag:=false;
    readln(schoollist);
    if (copy(schoollist,1,1)='') and
(copy(schoollist,length(schoollist),1)=='')//removing ""
    then schoollist:=copy(schoollist,2,length(schoollist)-2);
    if NOT(copy(schoollist,length(schoollist)-2,3)='txt') and
NOT(copy(schoollist,length(schoollist)-2,3)='TXT')//checking extension
    then begin
```



## Multiple-Choice Analysis Report

```
textcolor(3);  
writeln('The file required is not a TXT file');  
end  
else flag:=true;  
until flag;  
  
8 8888 .o8888888o.88888888 888888888888  
8 8888 8888 '88. 8 8888  
8 8888 ,8 8888 '8. 8 8888  
8 8888 88 8888 8 8888  
8 8888 88 8888 8 8888  
8 8888 88 8888 8 8888  
8 8888 88 8888 8 8888  
8 8888 '8 8888 .8 8 8888  
8 8888 8888 ,88 8 8888  
8 8888 '8888888P 8 8888      COMPETITION MC ANALYSIS  
*****  
Please input the file name of the participants' information and answers  
default:<MC DATA.txt> :  
The file required is not a TXT file  
Please input the file name of the participants' information and answers  
default:<MC DATA.txt> :HAHHA  
The file required is not a TXT file  
Please input the file name of the participants' information and answers  
default:<MC DATA.txt> :123.txt  
The file required to be opened is not found.  
  
Please input the file name of the participants' information and answers  
default:<MC DATA.txt> :C:\Users\admin\Desktop\Untitled.jpg  
The file required is not a TXT file  
Please input the file name of the participants' information and answers  
default:<MC DATA.txt> :MC DATA.txt  
Please input the file name of the answer key  
default:<ANS KEY.txt> :  
The file required is not a TXT file  
Please input the file name of the answer key  
default:<ANS KEY.txt> :HAHHA  
The file required is not a TXT file  
Please input the file name of the answer key  
default:<ANS KEY.txt> :123.txt  
The file required is not a TXT file  
Please input the file name of the answer key  
default:<ANS KEY.txt> :C:\Users\admin\Desktop\Untitled.jpg  
The file required is not a TXT file  
Please input the file name of the answer key  
default:<ANS KEY.txt> :ANS KEY.txt  
Please input the file name of school list  
default:<SCHOOL.txt> :  
The file required is not a TXT file  
Please input the file name of school list  
default:<SCHOOL.txt> :HAHHA  
The file required is not a TXT file  
Please input the file name of school list  
default:<SCHOOL.txt> :123.txt  
The file required to be opened is not found.  
  
Please input the file name of school list
```

## Multiple-Choice Analysis Report

```
default:<SCHOOL.txt> :SCHOOL.txt
LI KA SHING M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY
NG SIU YEE F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY
LEUNG HIU FUNG M SPRING SCHOOL
HOWARD YOUNG M FALL SECONDARY SCHOOL
SO PANG HIN M MOON SECONDARY SCHOOL
ADAM HENDERSON M VENUS COLLEGE
CHAN TIN WAI M WINTER SECONDARY SCHOOL
FONG CHEUK HO M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY
CHAN PUI HAN F MOON SECONDARY SCHOOL
CYNTHIA MARTIN F VENUS COLLEGE
BRENDA RIVERA F MARS COLLEGE
JANICE BROOKS F VENUS COLLEGE
TAM WING YAN F FALL SECONDARY SCHOOL
CARL ALEXANDER M MAPLE TREE COLLEGE
TERESA JENKINS F HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY
WONG NGA YIN BELINDA F EARTH COLLEGE
LEE SHU JEN M LEAGUE SCHOOL
LEUNG WAI CHI M EARTH COLLEGE
LUI KEI KWAN M LEAGUE SCHOOL
RAYMOND COX M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY
SENA KOBAYAKAWA M MAPLE TREE COLLEGE
MARILYN RUSSELL F VENUS COLLEGE
KATHY BAKER F SUN MEMORIAL SCHOOL
CHAN WING YEE F SPRING SCHOOL
AU WING HING M FALL SECONDARY SCHOOL
CHU KIN FUNG M WINTER SECONDARY SCHOOL
FONG KA WING M FALL SECONDARY SCHOOL
LAM MAN CHUNG M MAPLE TREE COLLEGE
LAW HO YU M FALL SECONDARY SCHOOL
MAR TIN PONG M SUN MEMORIAL SCHOOL
NG WAI CHUNG M WINTER SECONDARY SCHOOL
GARY WOOD M MAPLE TREE COLLEGE
TAYLOR WASHINGTON M WINTER SECONDARY SCHOOL
GREGORY LONG M MARS COLLEGE
LEUNG HOT YAN F VENUS COLLEGE
```

```
RONALD REED M LEAGUE SCHOOL
RALPH DAVIS M HOGWARTS SCHOOL OF WITCHCRAFT AND WIZARDRY
PAMELA WOOD F MOON SECONDARY SCHOOL
JANE JOHNSON F LEAGUE SCHOOL
PATRICIA KING F WINTER SECONDARY SCHOOL
BONNIE PHILLIPS F MAPLE TREE COLLEGE
ANGELA COLEMAN F EARTH COLLEGE
LEUNG CHEUK WING M LEAGUE SCHOOL
LI SI XIAN M LEAGUE SCHOOL
SO TING PONG M MARS COLLEGE
WILLIAM SANCHEZ M LEAGUE SCHOOL
HAROLD EVANS M SPRING SCHOOL
ANDREA EVANS F WINTER SECONDARY SCHOOL
WONG SAI MAN M SUMMER SCHOOL
CHU TSZ HIM M VENUS COLLEGE
```

Data successfully read  
Press <Enter> to enter the main menu

Now the program can both reject empty strings, file of non-text type and file that does not exist.

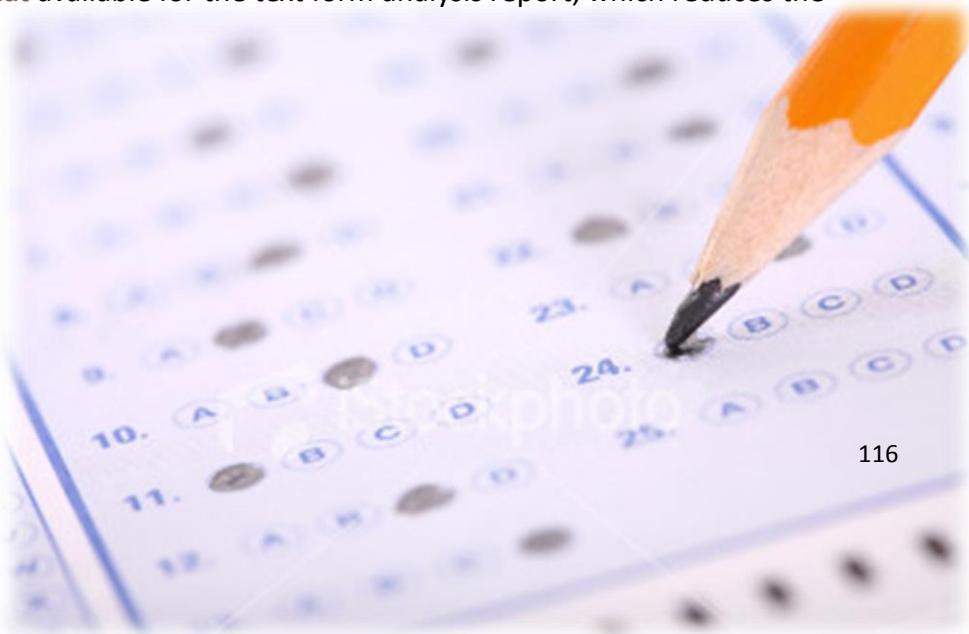
## 5.4 Self-Evaluation

Good features:

- ☺ The **welcoming screen**, **main menu** and the **ending screen** of the program are **vivid** as they have been **decorated with different ASCII art, colours and animation**, which users may find them **friendly**.
- ☺ The **instructions** and **words** are in **different colours** so that the user can identify whether certain words are inputted by them or not. It is rather **user-friendly**.
- ☺ The program **fulfills user's requirement** to generate a complete analysis report which contains different kinds of useful statistics.
- ☺ The analysis report can either be **on-screen** or a **portable text file**, which is **flexible** for the users
- ☺ The **layouts** for both on-screen or text form analysis report are **well-structured**. The user can easily locate the information.
- ☺ **Data validation** is performed **extensively**. All of the data inputted by the user will be checked by the program, which enables it to run smoothly.
- ☺ The coding is quite **readable** as the codes have been **indented** in a **neat manner**. Programmers may make **amendments** to the **codes** easily in the future.

Shortcomings:

- ☺ **Algorithms** used in this program are a compromise between efficiency and coding easiness. If the program is to handle a greater amount of data, the **efficiency will significantly decrease**.
- ☺ In the **main menu** of the program, the **options** are not written in **procedural forms**. Instead, they are only **compounded statements**. This may **confuse** the programmer when further improvements are made on the program.
- ☺ The analysis report on the **portable text file** is **dull** as formatting is not available in text file.
- ☺ There is only **one format** available for the text form analysis report, which reduces the program's **flexibility**.



## 5.5 External Testing and Evaluation

10 friends of mine have been asked to use the program and requested to give some comments. Here is a summary to the comments:

Positive:

- ☺ They all agree that the program is **user-friendly** as instructions are clear, the program can run **smoothly**.
- ☺ The use of ASCII art is quite **creative**.

Suggestions for improvement:

- If the program can have certain kind of **graphic interface**, the user will absolutely be more delighted to use the program as that will be much more **attractive**.

Response: Though graphic interface is very attractive, it is **out of my ability** to do so. It requires more advanced skills of programming. I will **try to improve** the program in this way **afterwards**.

- The analysis report is so **dull** that they are **not pleased** to read it through. If some charts are provided, that will be better.

Response: Though charts are really effective tools than just numbers, it is **out of my ability** to do so. It requires more advanced skills of programming that I can output the text file in MS Word form or HTML syntax. I will **try to improve** the program in this way **afterwards**.



# **Chapter 6 Conclusion & Discussion**

## **6.1 Pros and cons of the program**

Pros:

- ☺ The **welcoming screen**, **main menu** and the **ending screen** of the program are **vivid** as they have been **decorated with different ASCII art, colours** and **animation**, which users may find them **friendly**.
- ☺ The **instructions** and **words** are in **different colours** so that the user can identify whether certain words are inputted by them or not. It is rather **user-friendly**.
- ☺ The program **fulfills user's requirement** to generate a complete analysis report which contains different kinds of useful statistics.
- ☺ The analysis report can either be **on-screen** or a **portable text file**, which is **flexible** for the users
- ☺ The **layouts** for both on-screen or text form analysis report are **well-structured**. The user can easily locate the information.
- ☺ **Data validation** is performed **extensively**. All of the data inputted by the user will be checked by the program, which enables it to run smoothly.
- ☺ The coding is quite **readable** as the codes have been **indented** in a **neat manner**. Programmers may make **amendments** to the **codes** easily in the future.

Cons:

- ☺ **Algorithms** used in this program are a compromise between efficiency and coding easiness. If the program is to handle a greater amount of data, the **efficiency will significantly decrease**.
- ☺ In the **main menu** of the program, the **options** are not written in **procedural forms**. Instead, they are only **compounded statements**. This may **confuse** the programmer when further improvements are made on the program.
- ☺ The analysis report on the **portable text file** is **dull** as formatting is not available in text file.
- ☺ There is only **one format** available for the text form analysis report, which reduces the program's **flexibility**.

## 6.2 Future Improvement

As from [Chapter 5](#), after testing and evaluation, the pros and cons of the program have been found, in this section, some **suggestions for improvement** can be made according to that.

- ✿ **Graphic user interface (GUI)** can be introduced in the program to make the program more user-friendly, such that user can control the program just by clicking and drag and drop.
- ✿ The **use of charts** can be introduced to the analysis report to make the analysis more effective and readable.
- ✿ The user can choose **in what format** they like the **analysis report** to be like.
- ✿ The user can **assign scores** to specific **questions**.
- ✿ More **assessment-specific statistical measures** to the analysis can be added, like **item discrimination index** and **item difficulty index**, so that the analysis of the competition can be more effective.
- ✿ **Topics** of the question can be added to analyze which kind of topics that students are good at, so the way of making out questions can be altered in the competition next time.
- ✿ As the usual competition always have students divided into **several age groups**, the program can also provide analysis of students in specific age groups.
- ✿ **More efficient algorithms** can be used, like binary search for searching and quicksort for sorting.
- ✿ Program coding can be **neater** with better use of procedures in the main menu.



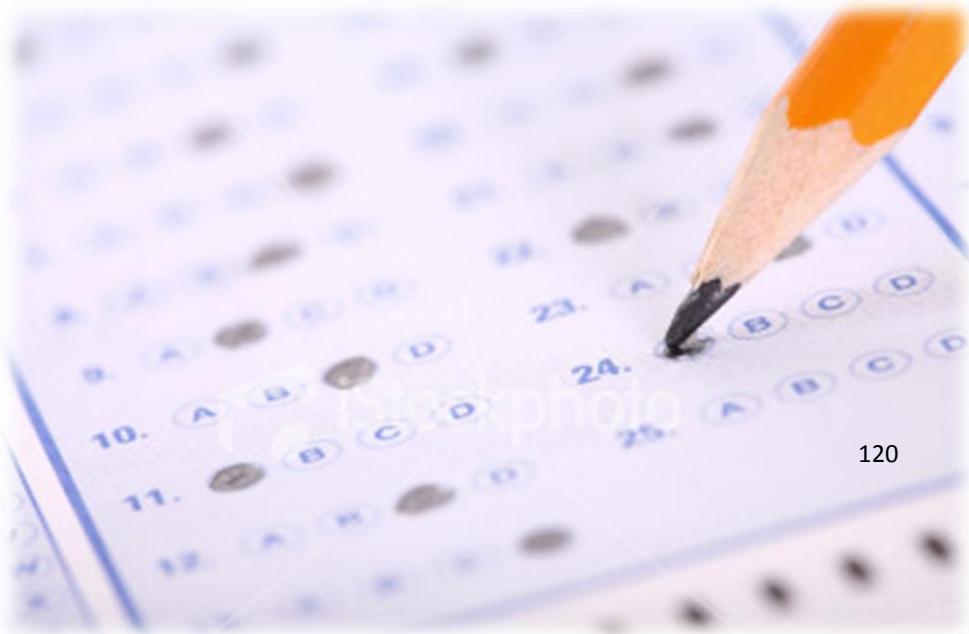
## 6.3 Self-Reflection

This is really an invaluable experience for me to have such project to accomplish. Although the process was tough, it is quite happy and satisfying to see the first program with over 900 lines written by my hands.

On working progress, I do realize the importance of planning and time management. The workload of this project is out of my expectation, the tasks I have been required to do is so enormous that I have delayed my working progress. A short program like this takes a week to be written, tested and debugged. If a programmer like me is to work on a 1000 times greater program like an OS system, if one does not strictly plan and follow the working plan, it would be disastrous. Therefore, it is really important when it comes to a chance for me to write a commercial program in society.

On learning process, I have learnt much more about programming. It is very true for people to say that it is better to practice than learning numerous concepts. In a normal ICT lesson, we seldom have chance to write a complete program due to a short period of time available in a lesson. Yet, this project enables students to practice all the things they learnt from lessons. From basic arithmetic operations to complicated data structures and different searching and sorting algorithms, they all got into practice. It definitely improved my programming knowledge and skills a lot, including the skills in designing algorithms and the sense to debug a program.

Also, it is not a cliché to say that in this new Information Age, it is very important for a student to learn in the Internet. Thanks to the Internet, I have searched much relevant knowledge on Pascal programming, where some of them are not taught in the text book.



# **Chapter 7 Reference and**

## **Acknowledgement**

### **Reference**

From Internet websites (listed in no particular order)

<http://www.nczonline.net/blog/wp-content/uploads/2012/09/insertionsort.png> (Illustration of insertion sort)

[http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Insertion\\_sort.html](http://www.princeton.edu/~achaney/tmve/wiki100k/docs/Insertion_sort.html) (Pros and cons of insertion sort)

[http://wiki.answers.com/Q/What\\_are\\_advantages\\_and\\_disadvantages\\_of\\_linear\\_search?#slide=5](http://wiki.answers.com/Q/What_are_advantages_and_disadvantages_of_linear_search?#slide=5) (Pros and cons of linear search)

[http://www.w3schools.com/browsers/browsers\\_os.asp](http://www.w3schools.com/browsers/browsers_os.asp) (OS Statistics)

<http://www.behindthename.com/random/> (Random Name Generator)

<http://patorjk.com/software/taag/#p=display&f=Graffiti&t=Type%20Something%20>

<http://pascal-programming.info/index.php> (Pascal Programming – Home Page)

<http://www.polyu.edu.hk/ssro/16-ssmsc/38-competition-format.html> (Inspiration to the criteria for individual winners)

<http://www.cprogramming.com/langs.html> (Pros and cons of different programming languages)

[http://en.wikipedia.org/wiki/Pascal\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Pascal_(programming_language)) (Introduction to Pascal)

[http://en.wikipedia.org/wiki/C\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/C_(programming_language)) (Introduction to C)

## Multiple-Choice Analysis Report

<http://en.wikipedia.org/wiki/C%2B%2B> (Introduction to C++)

[http://en.wikipedia.org/wiki/Visual\\_basic](http://en.wikipedia.org/wiki/Visual_basic) (Introduction to Visual Basic)

[http://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Java_(programming_language)) (Introduction to Java)

<http://en.wikipedia.org/wiki/Percentiles> (About percentiles)

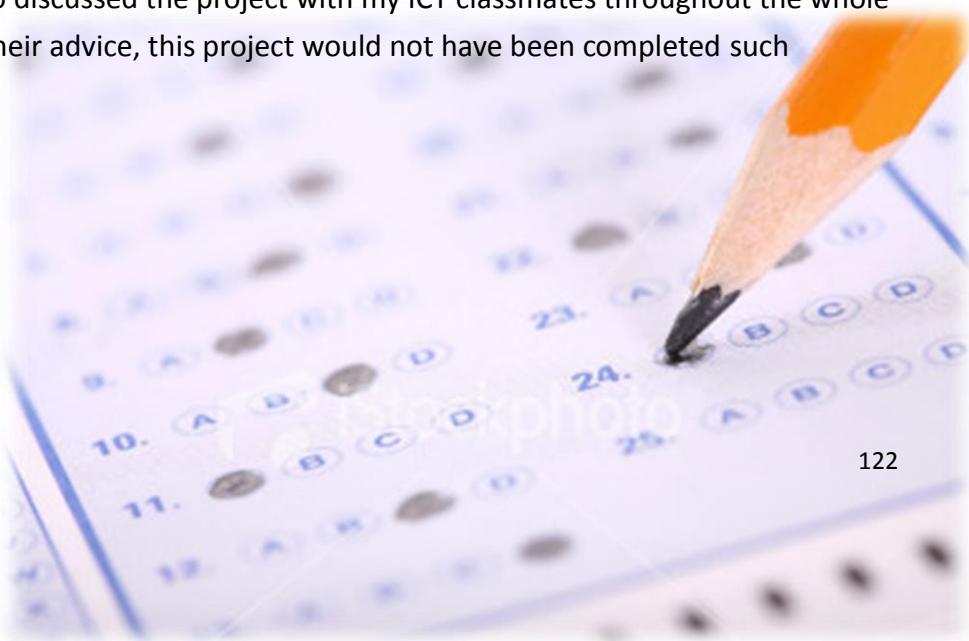
## From books (listed in no particular order)

1. Henry C H Ha, “NSS Information and Communication Technology Elective D1 Software Development” , *Longman Hong Kong Education*, Hong Kong, 2010.
- 2.
3. Henry C H Ha, “NSS Information and Communication Technology Elective D2 Software Development” , *Longman Hong Kong Education*, Hong Kong, 2010.
4. Kit Chan, “Computer and Information Technology for HKCEE-Module A2: Algorithm and Programming using Pascal”, *Digital Vision Educational Pub*, Hong Kong, 2003.

## Acknowledgement

I thank Mr. Chu Kin Fung, my ICT teacher, very much for his kind support to this project by giving enlightening instructions.

I also give credits to my own class 6A for them having their names in the list of sample participants’ file. I have also discussed the project with my ICT classmates throughout the whole working process, without their advice, this project would not have been completed such successfully.



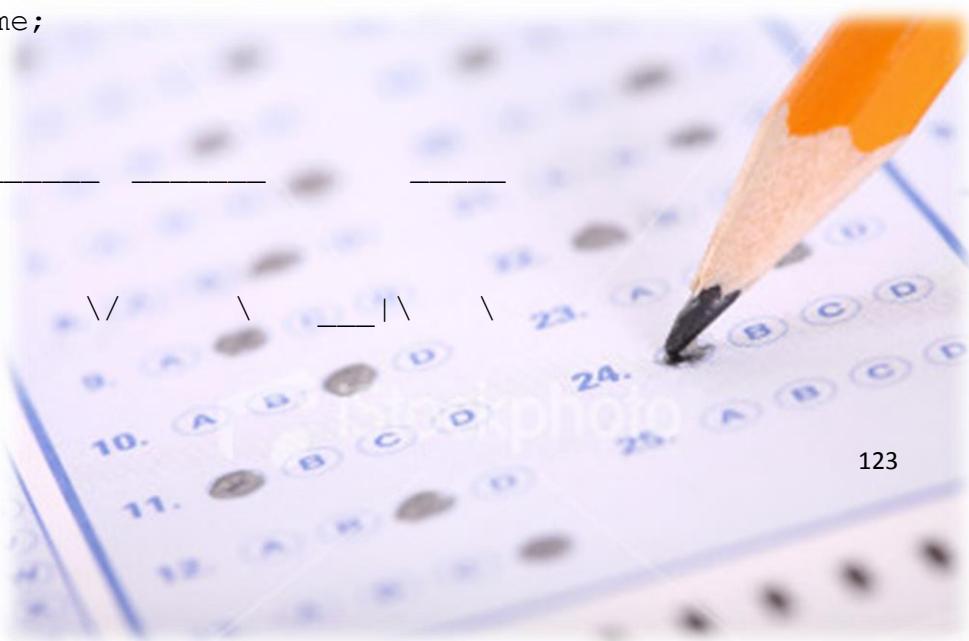
# Appendices

## Appendix I - Program Code

```
program ictmcanalysis;
uses crt;
const max=200;
var n,m,f,nos,tscore:integer;
    ans,schoollist : string;
    mean,median,d,SD:real;

    pans : array[1..max] of string[40];      //participants' answers
    pscore : array[1..max] of integer;        //participants' score
    pname : array[1..max] of string[25];      //participants' name
    sex : array[1..max] of char;              //participants' sex
    pschool : array[1..max] of string[45];    //participants' score
    school: array[1..999] of string[45];      //participating school
    snos: array[1..999] of integer;           //number of participants
in each school
    sscore:array[1..999] of integer;          //score of school
    qscore:array[1..40] of integer;            // number of correct in
each question
    percentcorrect:array[1..40] of real;      // %correct of each
question

{1}procedure welcome;
begin;
textcolor(14);
writeln('**');
delay(100);
writeln('**');
delay(100);
```



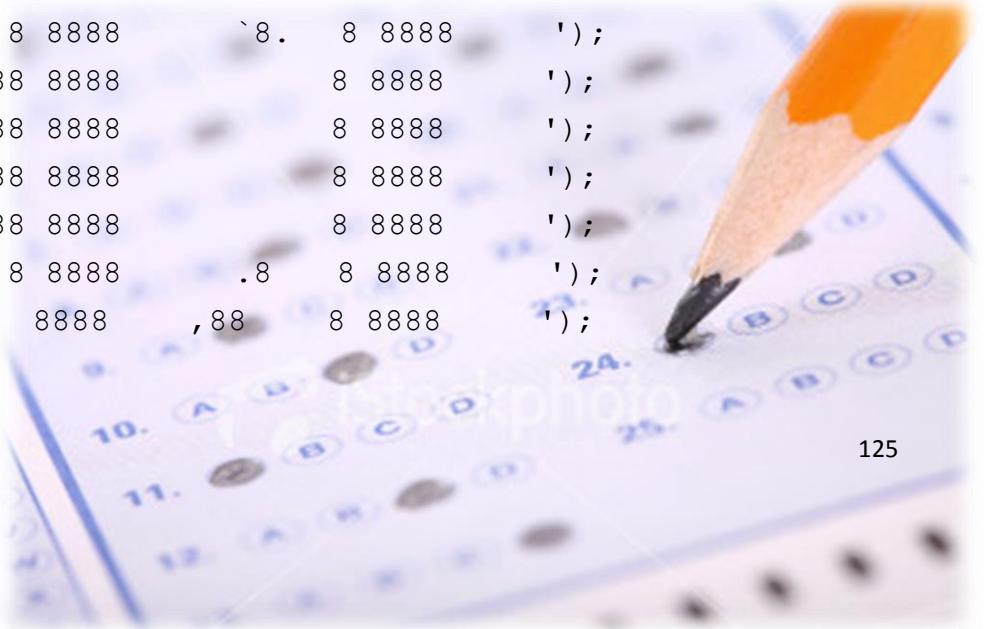
## Multiple-Choice Analysis Report

```
writeln('**      /      /\      \ /      /\\      \\  
**');  
delay(100);  
writeln('**      /      /\      / /      ||      | |      |  
**');  
delay(100);  
writeln('**      /      /\ \_/_/ / /      /||      | |_____|  
**');  
delay(100);  
writeln('**      |      | \_|/_| / /      / ||      | |_____|  
**');  
delay(100);  
writeln('**      |      |      |      | |      | |      | |  
**');  
delay(100);  
writeln('**      | \____\      |_____| / | \_ \ / / / |  
**');  
delay(100);  
writeln('**      | |      |      |      | / | |      /____/ |  
**');  
delay(100);  
writeln('**      \ |_____|      |_____| / \ |_____|      | /  
**');  
delay(100);  
writeln('**  
**');  
delay(100);  
writeln('**      _____ _      _____ -      _____ -  
_____ _____ * *');  
delay(100);  
writeln('**( ____ )( ( / | ( ____ )( \ | \ / / | ( ____ ) \ \_\_  
__ / ( ____ ) \ * *');  
delay(100);  
writeln('**| ( ) || \ ( | | ( ) || ( ( \ / ) | ( \ \ / ) ( |  
( \ / * *');  
delay(100);  
writeln('**| ( ____ ) || * \ | | ( ____ ) || | \ ( _ ) / | ( ____ ) | |  
| ( ____ * *');
```

## Multiple-Choice Analysis Report

```
delay(100);
writeln('**| ____ || (\ \ ) || ____ || | \ / (_____) || |
(_____) **');
delay(100);
writeln('**| ( ) || | \ || ( ) || | | ) ( ) | | | | | );
| **');
delay(100);
writeln('**| ) ( || ) \ || ) ( || (____/\| | / \____) | ____)
(____/\_____) | **');
delay(100);
writeln('**| / \|| / )_ ) | / \|| (_____/ \_ / \_____) \____ / \_____) *');
textcolor(3);
delay(500);
writeln('Welcome to the ICT competition multiple choice analysis
program.');
delay(1000);
write('Press <Enter> to continue');
readln;
end;
```

```
{2}procedure readinput;
const max=200;
var infile: text;
    pinfo,line: string;
    flag:boolean;
begin
    clrscr;
    writeln(' 8 8888 ,08888880.8888888 8888888888');
    writeln(' 8 8888 8888 `88. 8 8888 ');
    writeln(' 8 8888 ,8 8888 `8. 8 8888 ');
    writeln(' 8 8888 88 8888 8 8888 ');
    writeln(' 8 8888 88 8888 8 8888 ');
    writeln(' 8 8888 88 8888 8 8888 ');
    writeln(' 8 8888 `8 8888 .8 8 8888 ');
    writeln(' 8 8888 8888 ,88 8 8888 ');
```



## Multiple-Choice Analysis Report

```
writeln(' 8 8888      `8888888P      8 8888      COMPETITION MC
ANALYSIS');

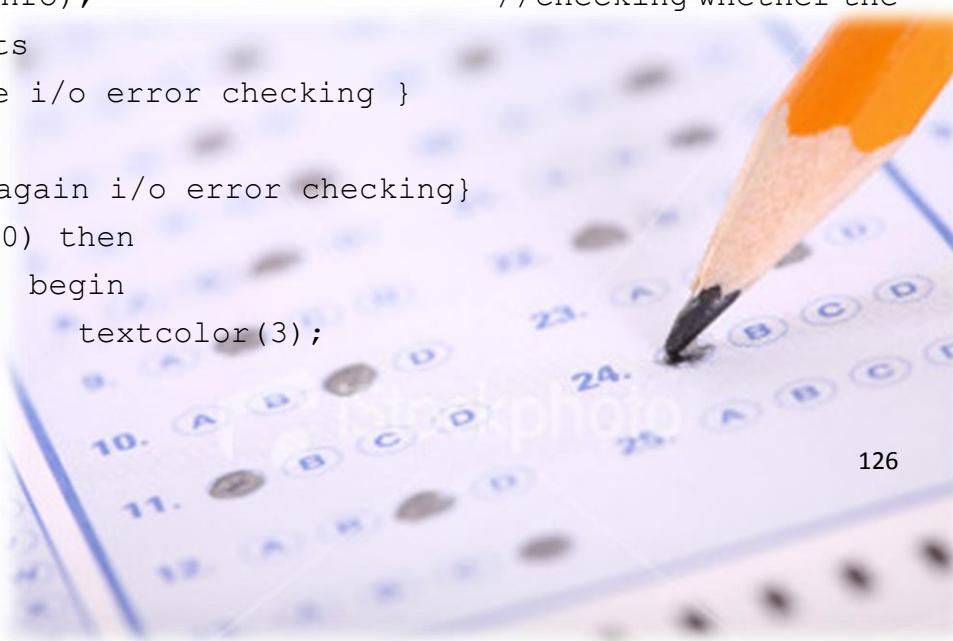
writeln('*****');
repeat
  repeat
    textColor(3);
    writeln('Please input the file name of the participants'''
information and answers');
    write('default:(MCDATA.txt) :');
    textColor(15);           {reading the participants' info}
    flag:=false;
    readln(pinfo);
    if (copy(pinfo,1,1)=""") and (copy(pinfo,length(pinfo),1)="""')
//removing ""
    then pinfo:=copy(pinfo,2,length(pinfo)-2);
    if NOT(copy(pinfo,length(pinfo)-2,3)='txt') and
NOT(copy(pinfo,length(pinfo)-2,3)='TXT') //checking extension
    then begin
      textColor(3);
      writeln('The file required is not a TXT file');

    end
    else flag:=true;

  until flag;

  flag:=false;
  assign(infile,pinfo);           //checking whether the
file inputted exists
  {$I-}  { disable i/o error checking }
  reset(infile);
  {$I+}  { enable again i/o error checking}
  if (IOResult <> 0) then
    begin
      textColor(3);

```



## Multiple-Choice Analysis Report

```
writeln('The file required to be opened is not
found.');
      readln;
    end
  else begin
    flag:=true ;
    close(infile);
  end

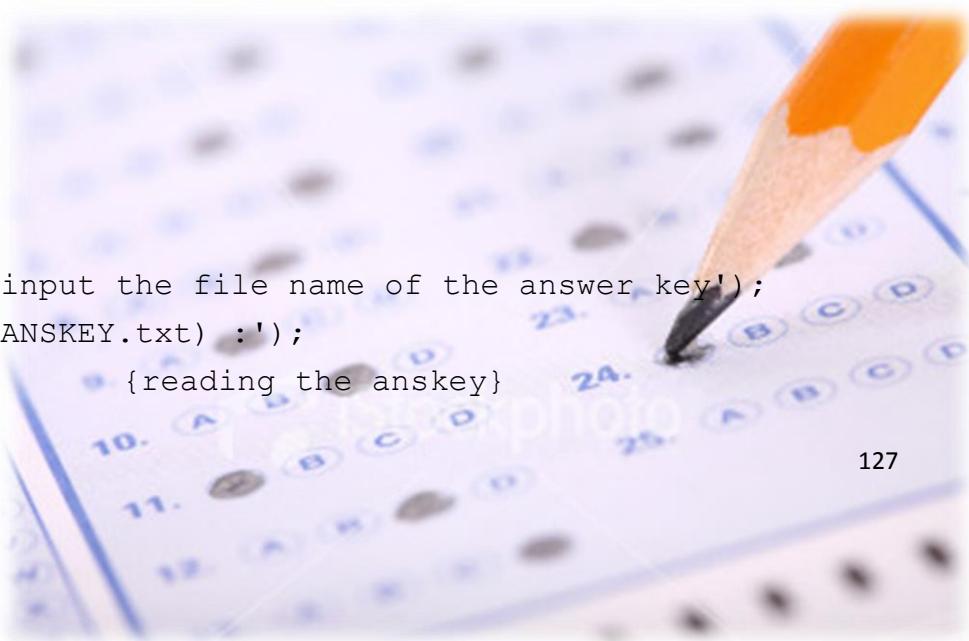
until flag=true;

assign(infile,pinfo);
reset(infile);
n:=0;

while not eof(infile) do begin
  n:=n+1;
  Readln(infile,line);
  pname[n]:=copy(line,1,25);
  sex[n]:=line[31];
  pschool[n]:=copy(line,33,45);
  pans[n]:=copy(line,83,40);
  if sex[n]='M' then m:=m+1; {counting male participants}
  if sex[n]='F' then f:=f+1; {counting female participants}

end;
close(infile); {end of reading the participants' info}
```

```
repeat
  repeat
    textColor(3);
    writeln('Please input the file name of the answer key');
    write('default:(ANSKEY.txt) :');
    textColor(15); {reading the anskey}
```



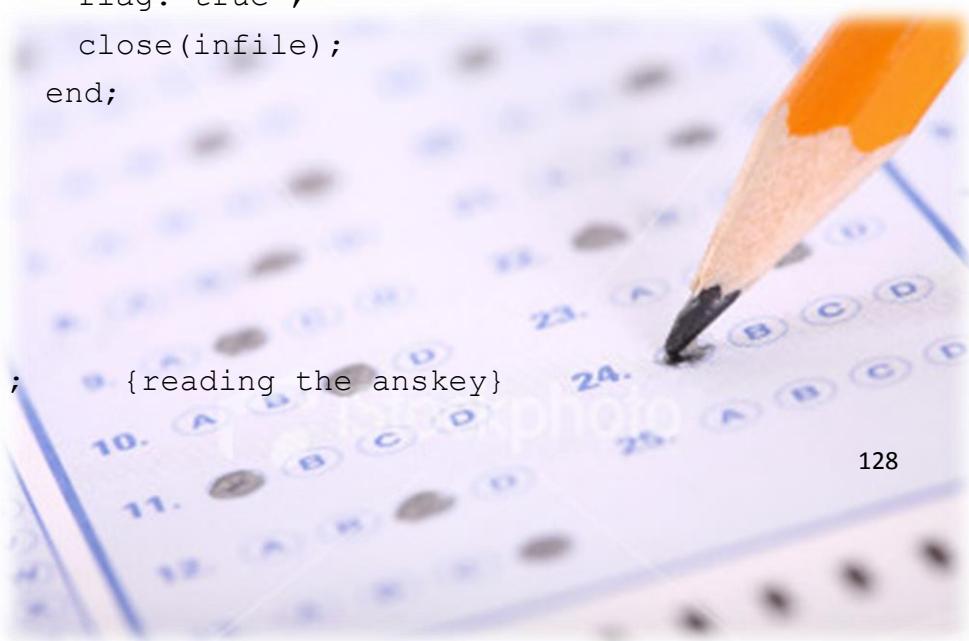
## Multiple-Choice Analysis Report

```
flag:=false;
readln(ans);
if (copy(ans,1,1)='') and (copy(ans,length(ans),1)='')
//removing ""
then ans:=copy(ans,2,length(ans)-2);
if NOT(copy(ans,length(pinfo)-2,3)='txt') and
NOT(copy(ans,length(ans)-2,3)='TXT')
then begin
    textcolor(3);
    writeln('The file required is not a TXT file');
    end
else flag:=true;

until flag;
flag:=false;

assign(infile,ans);
{$I-} { disable i/o error checking }
reset(infile);
{$I+} { enable again i/o error checking - important }
if (IOResult <> 0) then
begin
    textcolor(3);
    writeln('The file required to be opened is not
found.');
    readln;
end
else
begin
    flag:=true ;
    close(infile);
end;
until flag=true;

textcolor(15);
assign(infile,ans); {reading the anskey}
```

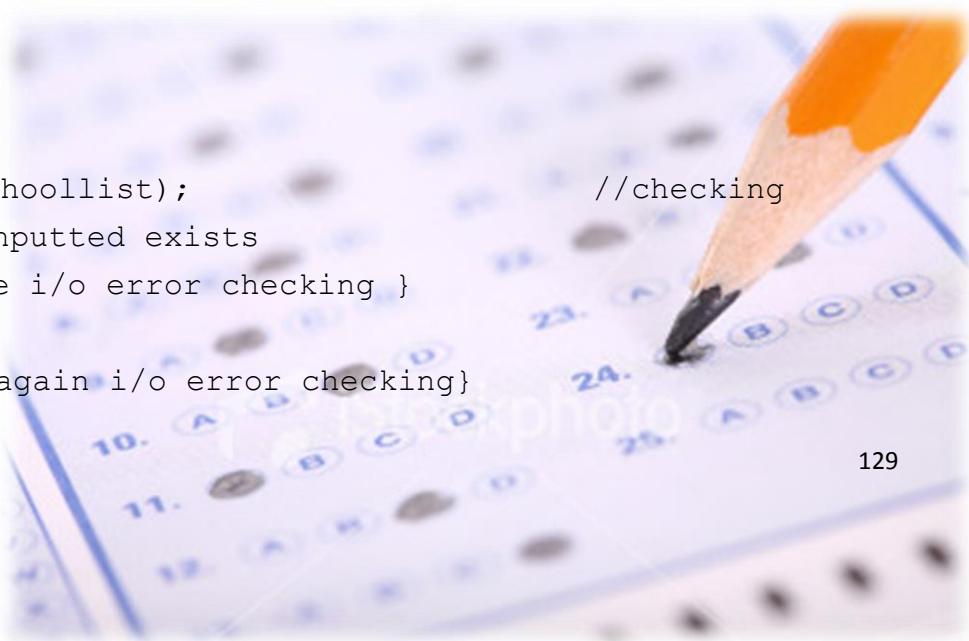


## Multiple-Choice Analysis Report

```
reset(infile);
readln(infile,ans);
close(infile);           {end of reading the anskey}

end;
{3}procedure readschool;
var  infile: text;
    line: string;
    i,j:integer;
    flag:boolean;
begin
repeat
repeat
textcolor(3);
writeln('Please input the file name of school list');
write('default:(SCHOOL.txt) :');
textcolor(15);           {reading the participants' info}
flag:=false;
readln(schoollist);
if (copy(schoollist,1,1)='') and
(copy(schoollist,length(schoollist),1)=='') //removing ""
then schoollist:=copy(schoollist,2,length(schoollist)-2);
if NOT(copy(schoollist,length(schoollist)-2,3)='txt') and
NOT(copy(schoollist,length(schoollist)-2,3)='TXT')
then begin
    textcolor(3);
    writeln('The file required is not a TXT file');
end
else flag:=true;

until flag;
flag:=false;
assign(infile,schoollist);           //checking
whether the file inputted exists
{$I-}  { disable i/o error checking }
reset(infile);
{$I+}  { enable again i/o error checking}
```



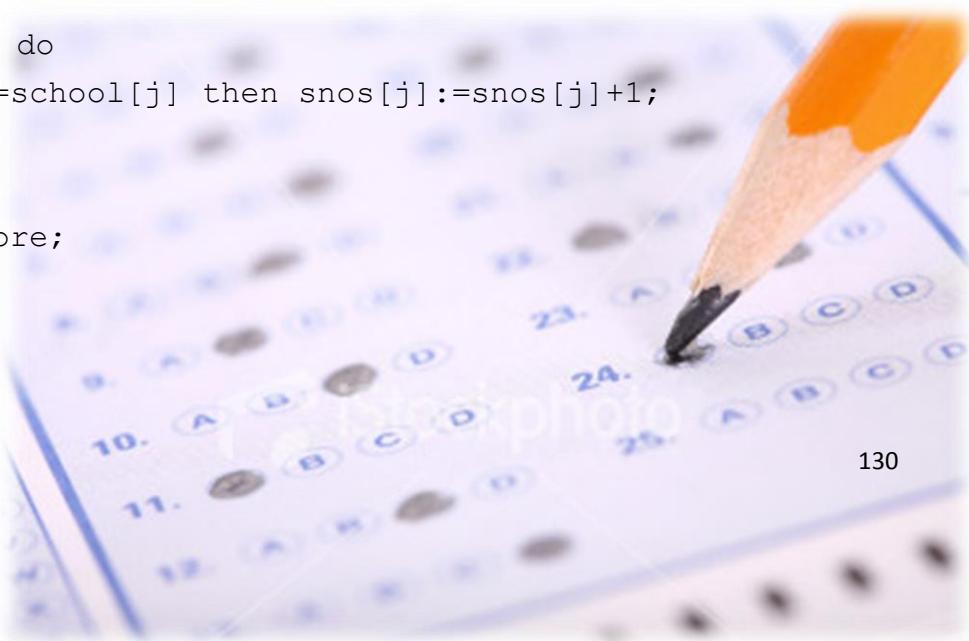
## Multiple-Choice Analysis Report

```
if (IOResult <> 0) then
    begin
        textColor(3);
        writeln('The file required to be opened is not
found.');
        readln;
    end
else
begin
    flag:=true ;
    close(infile);
end;
until flag=true;

textcolor(3);                                {reading the names of school}

textcolor(15);
assign(infile,schoollist);
reset(infile);
nos:=0;
while not eof(infile) do begin
    nos:=nos+1;                            {counting the number of participants in
each school}
    readln(infile,line);
    school[nos]:=copy(line,1,45);
end;
close(infile);                                {end of reading the names of school}

for i:= 1 to n do
begin
    for j:=1 to nos do
        if pschool[i]=school[j] then snos[j]:=snos[j]+1;
    end;
end;
{4}procedure calscore;
var
j,k,i: integer;
```



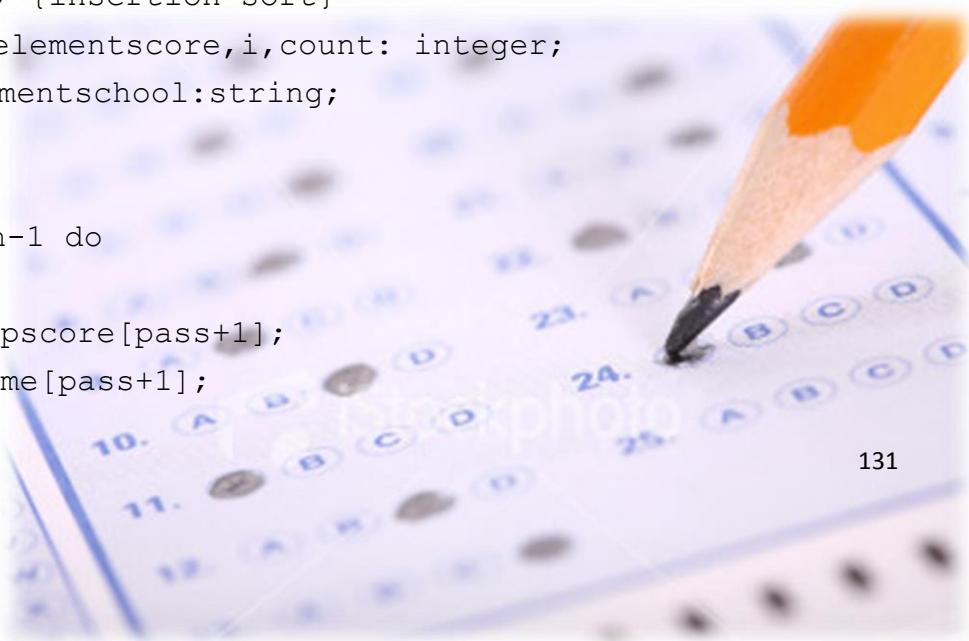
## Multiple-Choice Analysis Report

```
begin
    for j:= 1 to n do
        pscore[j]:=0;
    for j:= 1 to n do
        begin
            for k:= 1 to 40 do
                if (pans[j][k]=ans[k]) then begin
                    pscore[j]:=pscore[j]+1;
                    qscore[k]:=qscore[k]+1;
                end;
            end;
        for i:=1 to nos do
            sscore[i]:=0;
        for i:=1 to n do
            begin
                for j:=1 to nos do
                    if pschool[i]=school[j] then
                        sscore[j]:=sscore[j]+pscore[i];
                end;

            tscore:=0;
            for i:= 1 to n do
                tscore:=tscore+pscore[i];
            mean:=tscore/n;

        end;
```

```
{4}procedure sortp; {insertion sort}
Var pass,position,elementscore,i,count: integer;
    elementname,elementschool:string;
    elementc:char;
begin
    For pass := 1 to n-1 do
        Begin
            elementscore := pscore[pass+1];
            elementname:=pname[pass+1];
```



## Multiple-Choice Analysis Report

```
elementschool:=pschool[pass+1];
elementc:=sex[pass+1];
position:=1;
for i:= 1 to pass do
    if elementscore>pscore[i]
        then position:=i+1;

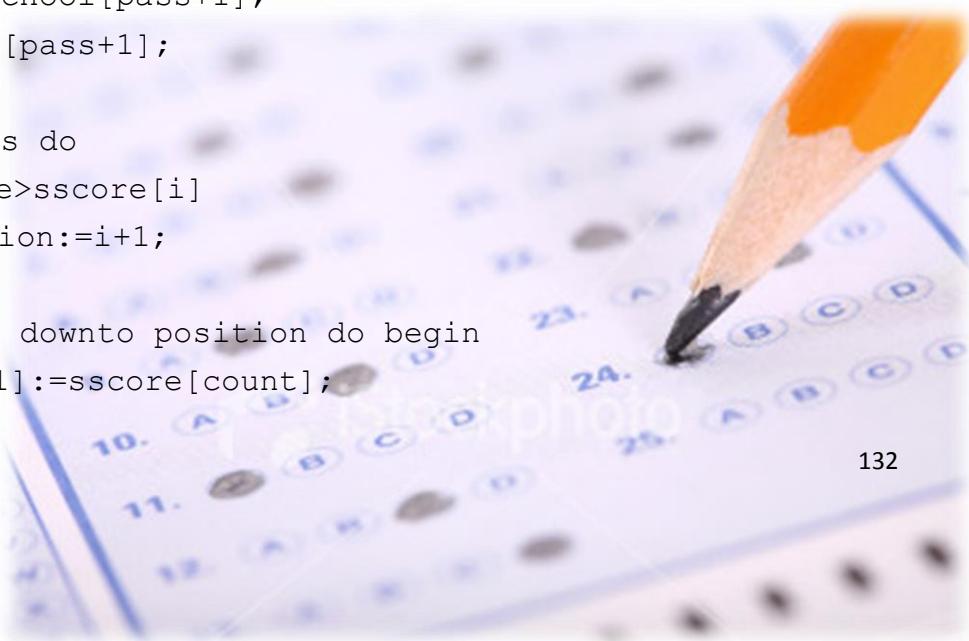
for count:= pass downto position do begin
    pscore[count+1]:=pscore[count];
    pname[count+1]:=pname[count];
    sex[count+1]:=sex[count];
    pschool[count+1]:=pschool[count];
end;

pscore[position]:=elementscore;
pname[position]:=elementname;
sex[position]:=elementc;
pschool[position]:=elementschool;

End;
end;

procedure sorts; {insertion sort}
Var pass,position,elementscore,elementnum,i,count: integer;
    elementschool:string;
begin
  For pass := 1 to nos-1 do
    Begin
      elementscore := sscore[pass+1];
      elementschool:=school[pass+1];
      elementnum:=snos[pass+1];
      position:=1;
      for i:= 1 to pass do
          if elementscore>sscore[i]
              then position:=i+1;

      for count:= pass downto position do begin
        sscore[count+1]:=sscore[count];
```



## Multiple-Choice Analysis Report

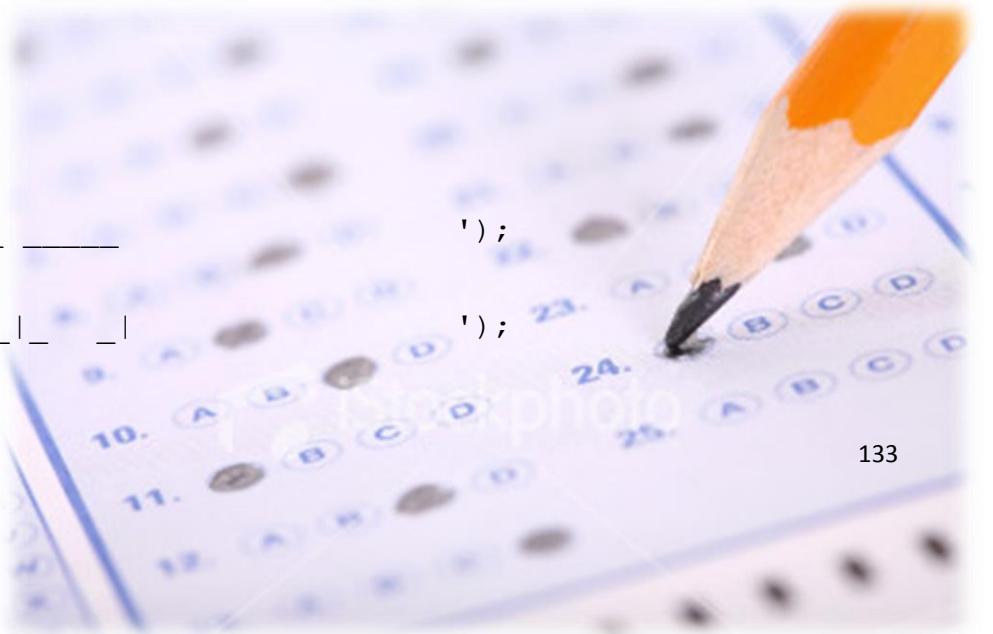
```
    school[count+1]:=school[count];
    snos[count+1]:=snos[count];
end;

    sscore[position]:=elementscore;
    school[position]:=elementschool;
    snos[position]:=elementnum;

End;
end;

{5}Procedure readdisplay;
var c:integer;
begin
    textColor(3);
    for c:=n downto 1 do
        writeln(pname[c],' ',sex[c],' ',pschool[c]);
    textColor(14);
    writeln('Data successfully read');
    write('Press <Enter> to enter the main menu');
    readln;
end;

{6}procedure mainmenu;
var q98,q85,q70,q50,q35,c,i,j,code,ioption,lenoption:integer;
    targetname,targetschool,txtfile,opt:string;
    targetsex:char;
    outfile:text;
    found:boolean;
begin
    clrscr;
    textColor(11);
    Gotoxy(20,1);
    writeln(' _____ ');
    Gotoxy(20,2);
    writeln(' |_ / ____|_ _| ');
    Gotoxy(20,3);
```



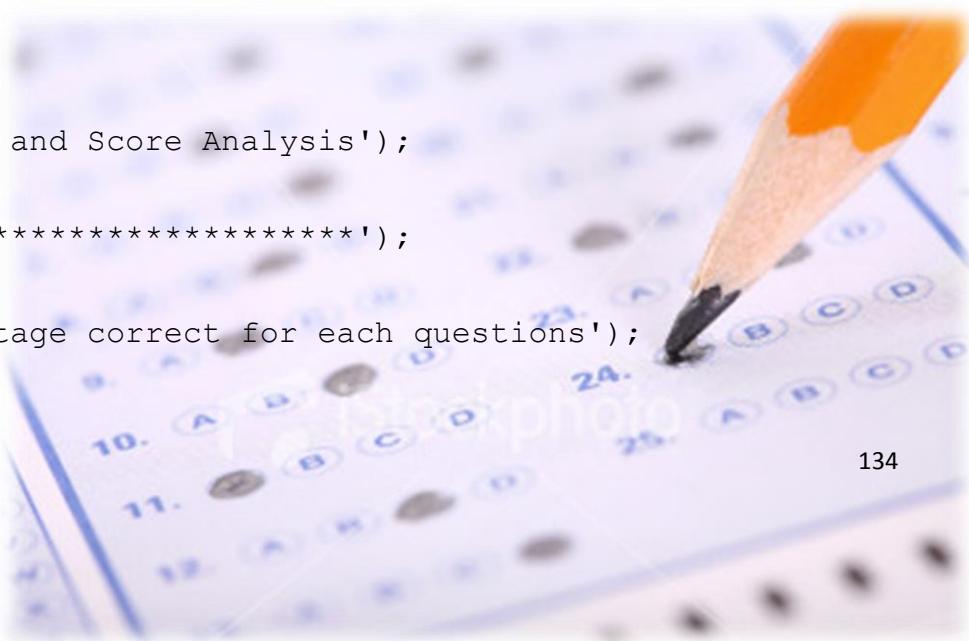
## Multiple-Choice Analysis Report

```
writeln(' | | |     | |      ') ;
Gotoxy(20,4) ;
writeln(' | | |_____| |      ') ;
Gotoxy(20,5) ;
writeln(' |____\____| |_| COMPETITION MC ANALYSIS') ;
writeln('*****') ;
writeln('*****') ;

textcolor(10) ;
gotoxy(5,8) ;
writeln('Numbers of participations') ;
gotoxy(5,9) ;
writeln('*****') ;
gotoxy(5,10) ;
writeln('1. Total number of participants') ;
gotoxy(5,11) ;
writeln('2. Total number of schools participated') ;
gotoxy(5,12) ;
writeln('3. Total number of participants from each schools') ;

textcolor(14) ;
gotoxy(58,8) ;
writeln('Awardees') ;
gotoxy(58,9) ;
writeln('*****') ;
gotoxy(58,10) ;
writeln('4. Individual awardees') ;
gotoxy(58,11) ;
writeln('5. School awardees') ;

textcolor(13) ;
gotoxy(5,14) ;
writeln('Questions and Score Analysis') ;
gotoxy(5,15) ;
writeln('*****') ;
gotoxy(5,16) ;
writeln('6. Percentage correct for each questions') ;
gotoxy(5,17) ;
```

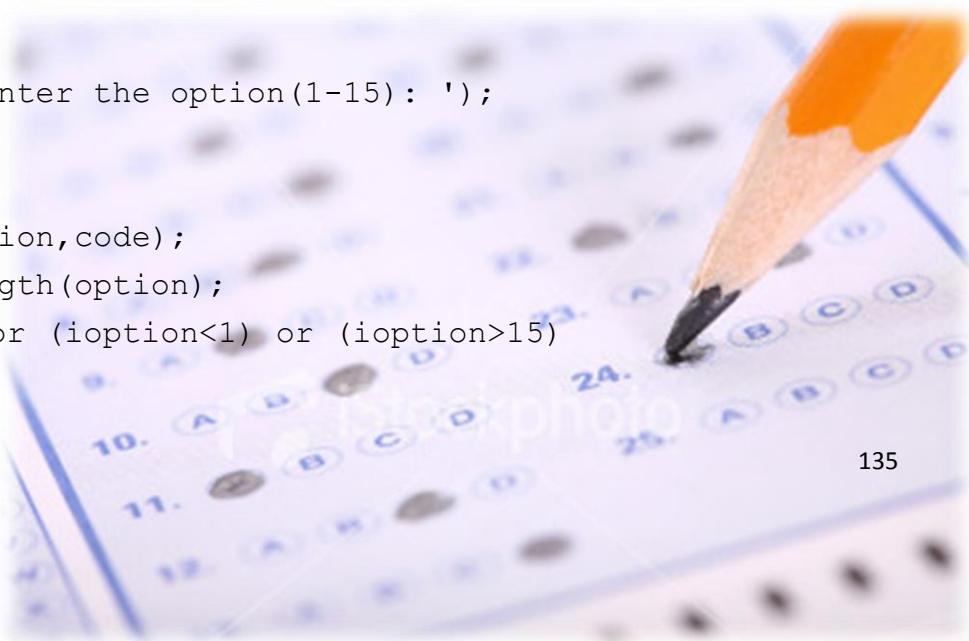


## Multiple-Choice Analysis Report

```
writeln('7. Standard deviation');
gotoxy(5,18);
writeln('8. Mean/ Median');
gotoxy(5,19);
writeln('9. Range/ Upper,lower-quartile/ Inter-quartile range');

textcolor(2);
gotoxy(58,14);
writeln('Search');
gotoxy(58,15);
writeln('*****');
gotoxy(58,16);
writeln('10. By name');
gotoxy(58,17);
writeln('11. By school');
gotoxy(58,18);
writeln('12. By sex');

textcolor(3);
gotoxy(27,23);
writeln('13. Display the list of all participants');
gotoxy(27,24);
writeln('14. Save report in text form');
gotoxy(27,25);
writeln('15. Quit');
c:=25;
textcolor(3);
gotoxy(27,c+1);
repeat
  write('Please enter the option(1-15): ');
  textcolor(15);
  readln(option);
  val(option,ioption,code);
  lenoption:= length(option);
  if (code<>0) or (ioption<1) or (ioption>15)
  then begin
```



## Multiple-Choice Analysis Report

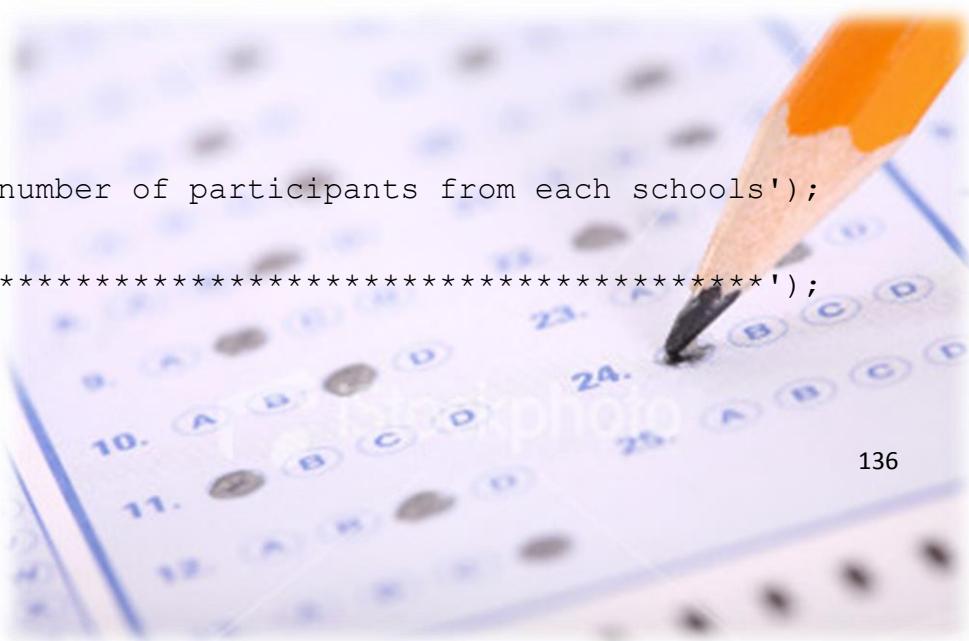
```
textcolor(3);
gotoxy(27,c+2);
writeln('Invalid input! Please try again.');
end;
c:=c+2;

until (ioption<=15) and (ioption>0);

case ioption of
1: begin
writeln;
textcolor(11);
writeln('Total number of participants for the competition: ',n);
writeln('Total number of male participants: ',m);
writeln('Total number of female participants: ',f);
write('Press <enter> to continue');
readln;
mainmenu;
end;

2: begin
writeln;
textcolor(11);
writeln('Total number of schools for the competition: ',nos);
write('Press <enter> to continue');
readln;
mainmenu;
end;

3: begin
clrscr;
Gotoxy(17,1);
textcolor(11);
writeln('Total number of participants from each schools');
Gotoxy(17,2);
writeln('*****');
Gotoxy(5,5);
```



## Multiple-Choice Analysis Report

```
writeln('School                                No. of
participants');

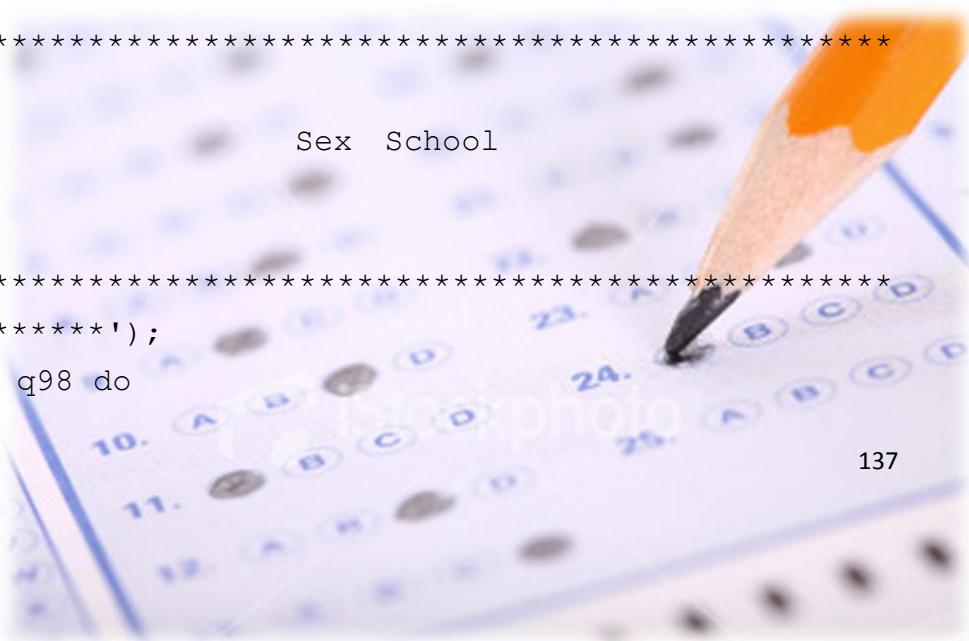
writeln('*****');
writeln('*****');
j:=7;
for i:=1 to nos do begin
  gotoxy(5,j);
  writeln(school[i],snos[i]);
  j:=j+1;
end;
writeln;
write('Press <enter> to continue');
readln;
mainmenu;
end;

4: begin
  q98:=round(98/100*n+0.5);
  q85:=round(85/100*n+0.5);
  q70:=round(70/100*n+0.5);
  q50:=round(50/100*n+0.5);
  q35:=round(35/100*n+0.5);
  clrscr;
  Gotoxy(15,1);
  textcolor(11);
  writeln('Here is the list of individual awardees of the
competition');
  Gotoxy(15,2);

writeln('*****');
writeln('*****');
writeln('Name          Sex  School
Award');

writeln('*****');
writeln('*****');

for i:=n downto q98 do
```



## Multiple-Choice Analysis Report

```
writeln(pname[i], ' ', sex[i], ' ', pschool[i], 'Medal');
for i:=q98-1 downto q85 do
  writeln(pname[i], ' ', sex[i], ' ', pschool[i], 'HD');
for i:=q85-1 downto q70 do
  writeln(pname[i], ' ', sex[i], ' ', pschool[i], 'D');
for i:=q70-1 downto q50 do
  writeln(pname[i], ' ', sex[i], ' ', pschool[i], 'Merit');
for i:=q50-1 downto q35 do
  writeln(pname[i], ' ', sex[i], ' ', pschool[i], 'Credit');
writeln;
writeln('Key: HD=High Distinction, D=Distinction');
writeln;
write('Press <enter> to continue');
readln;
mainmenu;
end;
5:begin
  clrscr;
  textcolor(11);

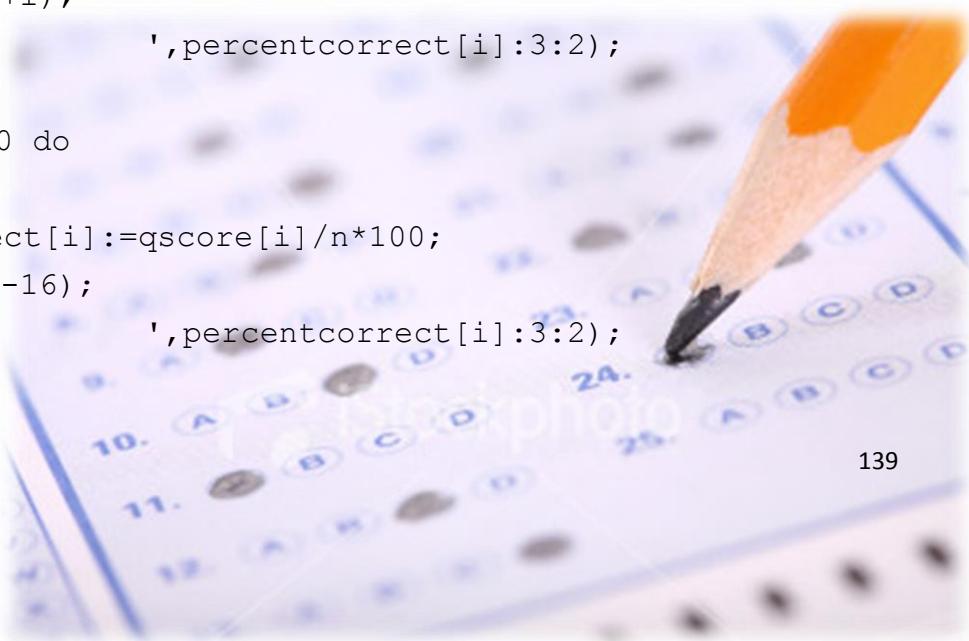
  Gotoxy(15,1);
  writeln('Here is the list of school award winners of the
competition');
  Gotoxy(15,2);

  writeln('*****');
  writeln('School           Total sum of score
Award');

  writeln('*****');
  writeln(school[nos], ' ', sscore[nos], '           Champion');
  writeln(school[nos-1], ' ', sscore[nos-1], '           1st
Runner Up');
  writeln(school[nos-2], ' ', sscore[nos-2], '           2nd
Runner Up');
  for i:=nos-3 downto 1 do
```

## Multiple-Choice Analysis Report

```
writeln(school[i], ' ', sscore[i], '  
Participation');  
writeln;  
write('Press <enter> to continue');  
readln;  
mainmenu;  
end;  
6:begin  
clrscr;  
textcolor(11);  
Gotoxy(15,1);  
writeln('Here is the list of percentage correct for each  
questions');  
Gotoxy(15,2);  
  
writeln('*****  
*');  
gotoxy(15,4);  
writeln('Question No. % correct Question No. %  
correct');  
for i:= 1 to 9 do  
begin  
percentcorrect[i]:=qscore[i]/n*100;  
gotoxy(21,4+i);  
writeln(i, ' ', percentcorrect[i]:3:2);  
end;  
for i:= 10 to 20 do  
begin  
percentcorrect[i]:=qscore[i]/n*100;  
gotoxy(21,4+i);  
writeln(i, ' ', percentcorrect[i]:3:2);  
end;  
for i:=21 to 40 do  
begin  
percentcorrect[i]:=qscore[i]/n*100;  
gotoxy(54,i-16);  
writeln(i, ' ', percentcorrect[i]:3:2);  
end;
```



## Multiple-Choice Analysis Report

```
writeln;
write('Press <enter> to continue');
readln;
mainmenu;
end;
7:begin
    textColor(11);

    d:=0;
    for i:=1 to n do
        d:=d+ (pscore[n]-mean) * (pscore[n]-mean);
    SD:=sqrt(d/n);

    writeln('Standard deviation of score is: ',SD:3:2);
    writeln;
    write('Press <enter> to continue');
    readln;
    mainmenu;
end;
8:begin
    textColor(11);
    writeln('Mean: ',mean:3:2);
    writeln('Median: ',pscore[round((1+n)/2)]);
    writeln;
    write('Press <enter> to continue');
    readln;
    mainmenu;
end;
9:begin
    textColor(11); {Range/
Upper,lower-quartile/ Inter-quartile range}
    writeln('Range: ',pscore[n]-pscore[1]);
    writeln('Upper quartile :',pscore[round(75/100*n+0.5)]);
    writeln('Lower quartile :',pscore[round(25/100*n+0.5)]);
    writeln('Inter-quartile
range :',pscore[round(75/100*n+0.5)]-pscore[round(25/100*n+0.5)]);
;
    writeln;
```

## Multiple-Choice Analysis Report

```
write('Press <enter> to continue');
readln;
mainmenu;
end;
10:begin
    writeln;
    textColor(3);
    writeln('Please enter the name of participant: ');
    textColor(15);           {search participant by name}
    readln(targetname);
    for i:=1 to length(targetname) do
        targetname[i]:=uppercase(targetname[i]);
    for i:=1 to (25-length(targetname)) do
        targetname:=targetname+' ';
    writeln(targetname);
    readln;
    clrscr;
    textColor(11);
    Gotoxy(28,1);
    writeln('Here is the record you want');
    gotoxy(28,2);
    writeln('*****');
    writeln;
    writeln('Name          Sex  School');
    writeln('Score');

writeln('*****');
writeln('*****');

found:=false;
for i:=n downto 1 do
    if targetname=pname[i] then begin
        writeln(pname[i], ' ', sex[i], ',',
        ',pschool[i],pscore[i]);
        found:=true;
    end;
    if found=false then writeln('RECORD NOT FOUND');
writeln;
```

## Multiple-Choice Analysis Report

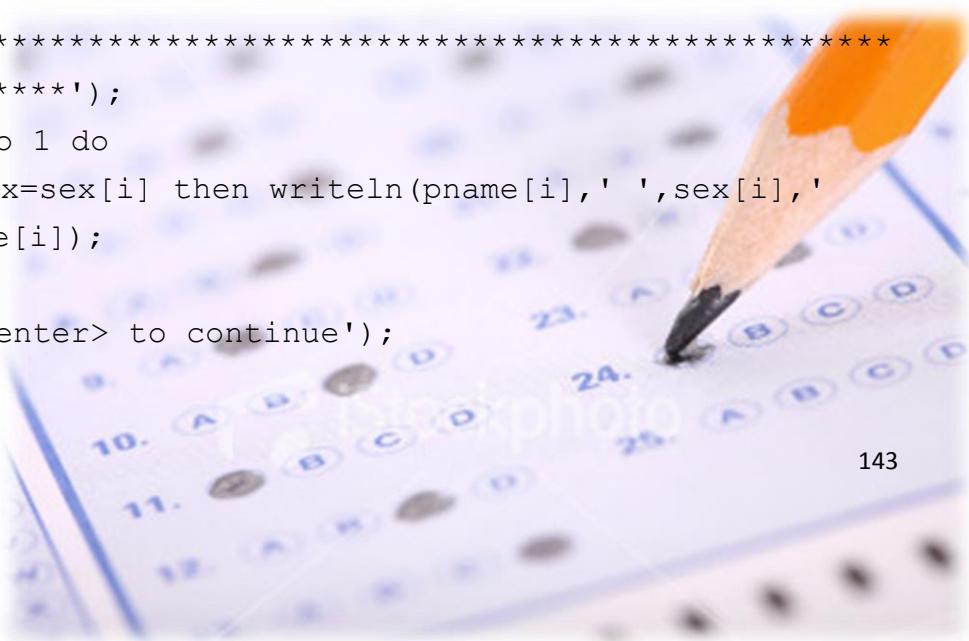
```
write('Press <enter> to continue');
readln;
mainmenu;
end;

11:begin
writeln;
textcolor(3);
writeln('Please enter the school of participant: ');
textcolor(15); {search participant by school}
readln(targetschool);
for i:=1 to length(targetschool) do
  targetname[i]:=upcase(targetschool[i]);
for i:=1 to (45-length(targetschool))do
  targetschool:= targetschool+' ';
writeln(length(targetschool));
readln;
clrscr;
textcolor(11);
Gotoxy(28,1);
writeln('Here is/are the record(s) you want');
gotoxy(28,2);
writeln('*****');
writeln;
writeln('Name           Sex   School
Score');

writeln('*****');
found:=false;
for i:=n downto 1 do
  if targetschool=pschool[i] then begin
    writeln(pname[i], ' ', sex[i], '
', pschool[i], pscore[i]);
    found:=true;
  end;
  if found=false then writeln('RECORD NOT FOUND');
writeln;
```

## Multiple-Choice Analysis Report

```
write('Press <enter> to continue');
readln;
mainmenu;
end;
12:begin
writeln;
repeat
  textColor(3);
  writeln('Please enter the sex of participant(M/F): ');
  textColor(15);                      {search participant by
name}
  readln(targetsex);
  if (targetsex<>'M') or (targetsex<>'F') or (targetsex<>'m') or
(targetsex<>'f') then begin
    textColor(3);
    writeln('Invalid
input, please enter M/F only');
    end;
  until (targetsex='M') or (targetsex='F') or (targetsex='m') or
(targetsex='f');
clrscr;
textcolor(11);
Gotoxy(28,1);
writeln('Here are the records you want');
gotoxy(28,2);
writeln('*****');
writeln;
writeln('Name          Sex  School
Score');
writeln('*****');
for i:=n downto 1 do
  if targetsex=sex[i] then writeln(fname[i], ' ', sex[i],
',', pschool[i], pscore[i]);
writeln;
write('Press <enter> to continue');
readln;
```



## Multiple-Choice Analysis Report

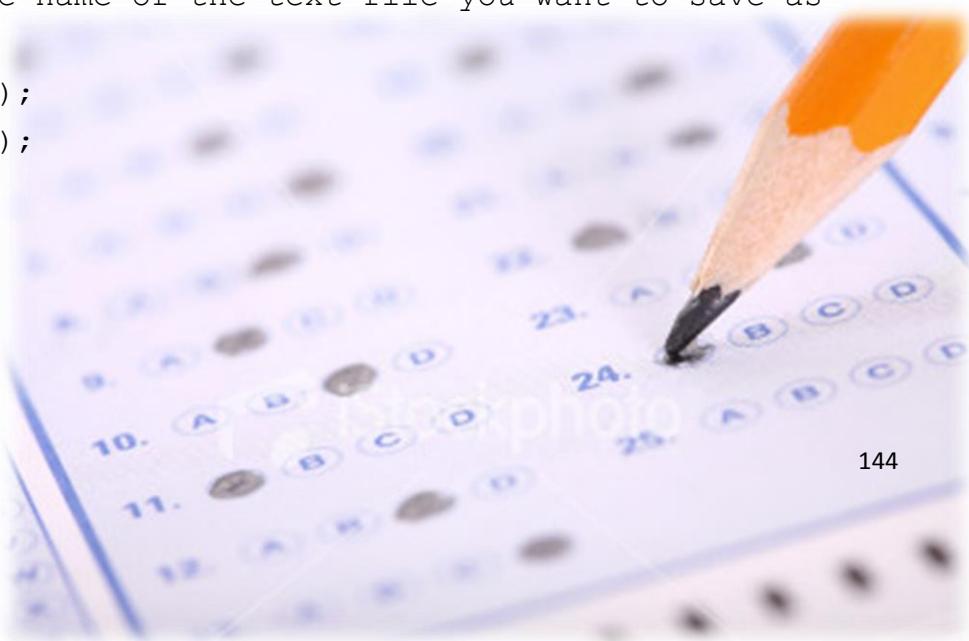
```
mainmenu;
end;

13:begin
    clrscr;                                { show all
participants}
    textColor(11);
    Gotoxy(20,1);
    writeln('Here is the list of all participants');
    Gotoxy(20,2);
    writeln('*****');
    writeln;

    writeln('Name          Sex  School
Score');

writeln('*****');
for i:=n downto 1 do
    writeln(pname[i],' ',sex[i],' ',pschool[i],pscore[i]);
writeln;
write('Press <enter> to continue');
readln;
mainmenu;
end;

14:begin
    textColor(11);
    writeln;
    write('Enter the name of the text file you want to save as
(with .txt): ');
    readln(textfile);
    write('Loading');
    delay(1000);
    write('.');
    delay(1000);
    write('.');
    delay(1000);
```



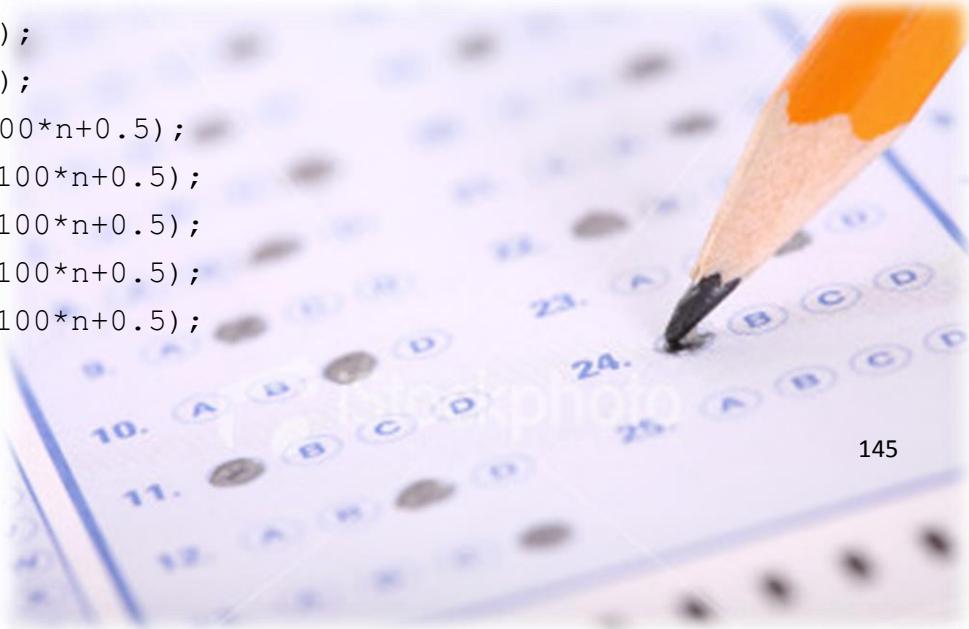
## Multiple-Choice Analysis Report

```
write('.');
writeln('Success');
assign(outfile,textfile);
rewrite(outfile);
writeln(outfile,'A Report on the ICT competition');
writeln(outfile,'-----');
writeln(outfile);
writeln(outfile,'Participation:');
writeln(outfile,'*****');
writeln(outfile,'Total number of participants for the competition:
',n);
writeln(outfile,'Total number of male participants: ',m);
writeln(outfile,'Total number of female participants: ',f);
writeln(outfile,'Total number of schools for the competition:
',nos);
writeln(outfile);
writeln(outfile);
writeln(outfile,'Total number of participants from each
schools');

writeln(outfile,'*****');
;

writeln(outfile,'School No. of
participants');

writeln(outfile,'*****');
for i:=1 to nos do begin
    writeln(outfile,school[i],snos[i]);
end;
writeln(outfile);
writeln(outfile);
q98:=round(98/100*n+0.5);
q85:=round(85/100*n+0.5);
q70:=round(70/100*n+0.5);
q50:=round(50/100*n+0.5);
q35:=round(35/100*n+0.5);
```

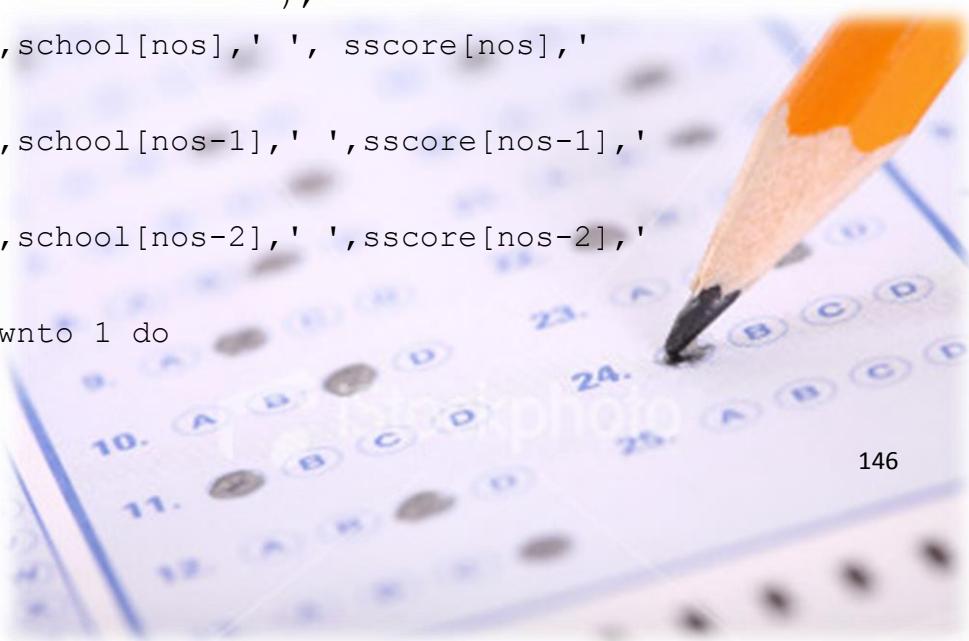


## Multiple-Choice Analysis Report

```
writeln(outfile,'AWARDS:');
writeln(outfile);
writeln(outfile,'Individual awardees of the competition');
writeln(outfile,'*****');
writeln(outfile,'Name           Sex   School');
writeln(outfile,'Award');

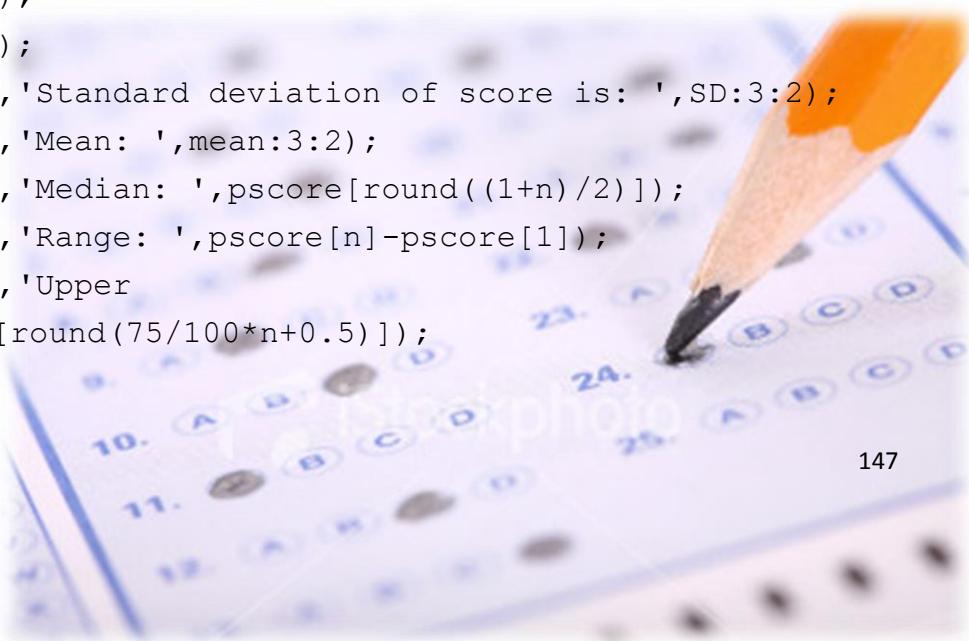
writeln(outfile,'*****');
for i:=n downto q98 do
  writeln(outfile, pname[i], ' ', sex[i], ' ', pschool[i], 'Medal');
for i:=q98-1 downto q85 do
  writeln(outfile, pname[i], ' ', sex[i], ' ', pschool[i], 'HD');
for i:=q85-1 downto q70 do
  writeln(outfile, pname[i], ' ', sex[i], ' ', pschool[i], 'D');
for i:=q70-1 downto q50 do
  writeln(outfile, pname[i], ' ', sex[i], ' ', pschool[i], 'Merit');
for i:=q50-1 downto q35 do
  writeln(outfile, pname[i], ' ', sex[i], ' ', pschool[i], 'Credit');
writeln;
writeln(outfile,'Key: HD=High Distinction, D=Distinction');
writeln(outfile);
writeln(outfile,'School award winners of the competition');
writeln(outfile,'*****');
writeln(outfile,'School           Total sum');
writeln(outfile,'of score      Award');

writeln(outfile,'*****');
writeln(outfile,school[nos], ' ', sscore[nos], ' ');
writeln(outfile,'Champion');
writeln(outfile,school[nos-1], ' ', sscore[nos-1], ' ');
writeln(outfile,'1st Runner Up');
writeln(outfile,school[nos-2], ' ', sscore[nos-2], ' ');
writeln(outfile,'2nd Runner Up');
for i:=nos-3 downto 1 do
```



## Multiple-Choice Analysis Report

```
writeln(outfile,school[i],' ',sscore[i],'  
Participation');  
writeln(outfile);  
writeln(outfile,'QUESTIONS AND SCORE ANALYSIS:');  
writeln(outfile);  
writeln(outfile,'Percentage correct for each questions');  
  
writeln(outfile,'*****');  
writeln(outfile,'*****');  
writeln(outfile,'Question No. % correct Question No.  
% correct');  
for i:=1 to 40 do  
    percentcorrect[i]:=qscore[i]/n*100;  
i:=1;  
for i:=1 to 9 do  
writeln(outfile,' ',i,' ',percentcorrect[i]:3:2,',  
,i+20,' ',percentcorrect[i+20]:3:2);  
for i:=10 to 20 do  
writeln(outfile,' ',i,' ',percentcorrect[i]:3:2,',  
,i+20,' ',percentcorrect[i+20]:3:2);  
  
d:=0;  
for i:=1 to n do  
d:=d+ (pscore[n]-mean) * (pscore[n]-mean);  
SD:=sqrt(d/n);  
writeln(outfile);  
writeln(outfile);  
writeln(outfile,'Standard deviation of score is: ',SD:3:2);  
writeln(outfile,'Mean: ',mean:3:2);  
writeln(outfile,'Median: ',pscore[round((1+n)/2)]);  
writeln(outfile,'Range: ',pscore[n]-pscore[1]);  
writeln(outfile,'Upper  
quartile :',pscore[round(75/100*n+0.5)]));
```



## Multiple-Choice Analysis Report

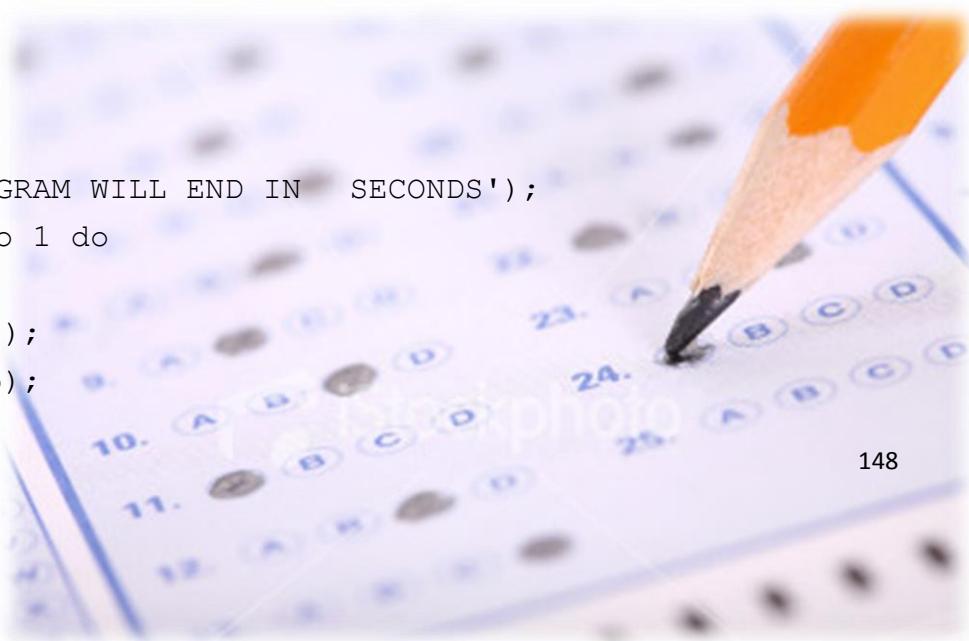
```
writeln(outfile,'Lower
quartile :',pscore[round(25/100*n+0.5)]);
writeln(outfile,'Inter-quartile
range :',pscore[round(75/100*n+0.5)]-pscore[round(25/100*n+0.5)])
;
writeln(outfile);
writeln(outfile);
writeln(outfile);
writeln(outfile);
writeln(outfile,'The list of all participants');
writeln(outfile,'*****');
writeln(outfile);

writeln(outfile,'Name           Sex   School
Score');

writeln(outfile,'*****');
for i:=n downto 1 do
writeln(outfile, pname[i], ' ', sex[i], '
', pschool[i], pscore[i]);

write('Press <enter> to continue');
readln;
close(outfile);
mainmenu;
end;

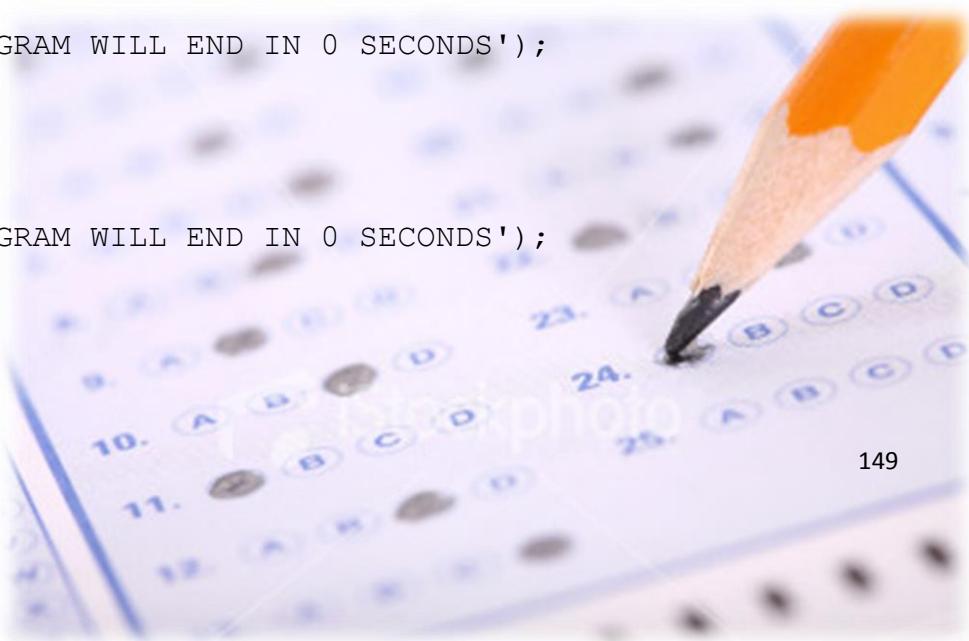
15:begin
clrscr;
textcolor(11);
gotoxy(25,15);
write('THE PROGRAM WILL END IN    SECONDS');
for i:=5 downto 1 do
begin
textcolor(14);
gotoxy(49,15);
```



## Multiple-Choice Analysis Report

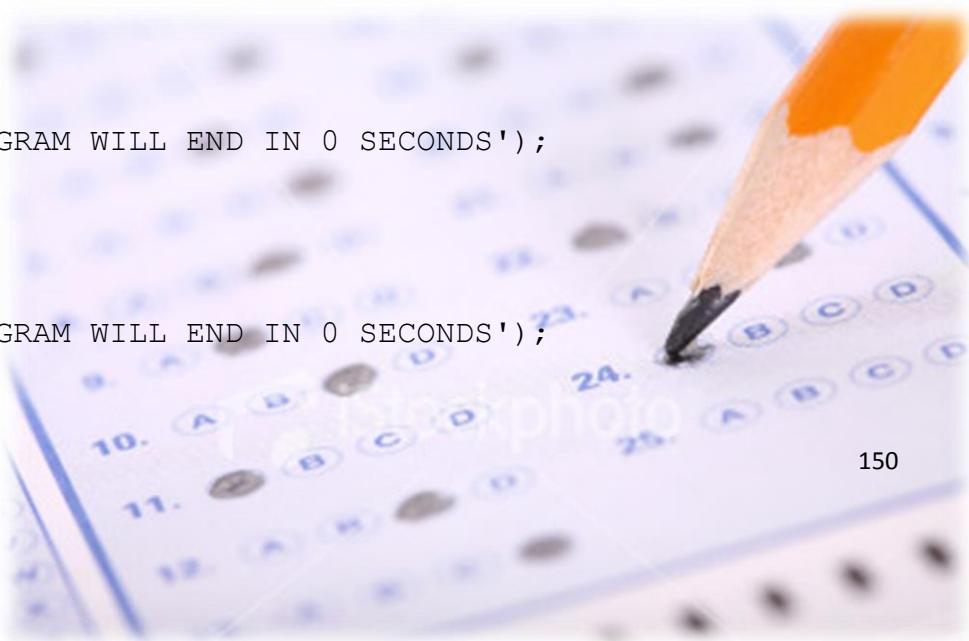
```
write(i);
delay(1000);
end;

gotoxy(25,15);
textcolor(14);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(11);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(14);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(11);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(14);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(11);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(14);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(11);
write('THE PROGRAM WILL END IN 0 SECONDS');
```



## Multiple-Choice Analysis Report

```
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(11);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(14);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(11);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(14);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(11);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(14);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(11);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(14);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(11);
write('THE PROGRAM WILL END IN 0 SECONDS');
delay(100);
gotoxy(25,15);
textcolor(14);
write('THE PROGRAM WILL END IN 0 SECONDS');
```



## Multiple-Choice Analysis Report

```
halt;  
end;  
end;  
end;  
  
begin      {main program}  
welcome;  
readinput;  
readschool;  
calscore;  
sorts;  
sortp;  
readdisplay;  
mainmenu;  
readln;  
end.
```



## Appendix 2 - Working schedule

Time/Date	Event
June 2013	Chapter 1 of the report : - + Planning of the whole working schedule
27-29/12	Chapter 2 and 3 : - Analysis of the system - Rough and general design of the system
30/12-4/1	Coding and general structure of the program completed
5/1-8/1	Chapter 4 : - Explaining the algorithms and justifying the program structures in the report - Amendments made to the program
9/1-11/1	Chapter 5 : - Testing evaluation and debugging of the program - The program is given to external users for testing, comments received - Optimise coding of the program Chapter 6 : Conclusion of the report, adding references
12/1-13/1	Finalise all the materials, formatting and indentation of documents

