Candidate Name: _____ CT Group: _____

Index no. _____

PIONEER JUNIOR COLLEGE
JC 2 PRELIMINARY EXAMINATION

H2 COMPUTING **9597/01**

Monday **16 SEPTEMBER 2013** **3 hours 15 min**

TIME 0800 - 1115

Additional Materials: Electronic version of **`RESULTS.txt`** data file
Electronic version of **`TRANSACTION.txt`** data file
Electronic version of **`MASTERFILE.txt`** data file
Electronic version of **`EVIDENCE.docx`** file

READ THESE INSTRUCTIONS FIRST

Type in the **`EVIDENCE.docx`** document the following:

- Candidate details (Name, Index No., Class)
- Programming language used

Answer **all** questions.

All tasks must be done in the computer laboratory. You are not allowed to bring in or take out any pieces of work or materials on paper or electronic media or in any other form.

All tasks and required evidence are numbered.

The number of marks is given in brackets [ ] at the end of each task.

Copy and paste required evidence of program listing and screen shots into the **`EVIDENCE.docx`** document.

At the end of the examination, print out your `EVIDENCE.docx` and fasten your printed copy securely together.

For Official Use:

	Question 1	Question 2	Question 3	Question 4	Total
Marks	**/ 20**	**/ 20**	**/ 35**	**/ 25**	**/ 100**

This question paper consists of **5** printed pages (inclusive of this page).

1 A program is to process the results of a competition, in which **eight** teams participated.

This competition consists of **five** rounds. For each round, the fastest finisher is ranked 1, the second fastest 2, and so on. No team should have the same rank in each round.

The first **four** rounds are already completed. A text file **`RESULTS.txt`** contains the team names and the average rankings of the first four rounds.

Example 1: If Team A finish 1st in each of the four rounds respectively, then it would have an average ranking of **1.0**.

Example 2: If Team B finish 1st, 2nd, 3rd, and 4th in rounds 1, 2, 3 and 4 respectively, then it would have an average ranking of (1+2+3+4) ÷ 4 = **2.5**.

The program specification is to

- display the average rankings of the 8 teams in the first four rounds
- input rankings for each of the 8 teams for the last round of competition
- calculate and display on the screen:
    - the updated average rankings of all teams
    - the winning team name and average ranking
- update **`RESULTS.txt`** with the final team names and average rankings

Task 1.1

Write program code for this task.

Evidence 1: Your program code. **[14]**

Task 1.2

Draw up a set of test data which tests the functioning of your program.

Evidence 2: A screenshot for each test case you considered. Annotate the screenshot explaining the purpose of each test. **[6]**

2 The following is a pseudocode algorithm to determine if a positive integer N is prime.

The algorithm is both poorly designed and contains errors.

```
FOR x = 1 TO N
     IF (N mod x) IS EQUAL TO 0
          flag = False
     ELSE
          flag = True
     ENDIF
ENDFOR
```

Task 2.1

Write program code for this algorithm including all the improvements you would make to:

- correct the errors
- follow good programming practice
- make the algorithm more efficient

Test your program with suitable positive integers.

Evidence 3: Your program code. **[9]**

Evidence 4: Four screenshots showing the output from running the program. **[4]**

Task 2.2

The prime number determination code could be useful in many real life applications, for example hashing functions, cryptography, etc. Re-design the program code to have a Boolean function **`IsPrime`**. This function should have a parameter which allows it to be used for any positive integer. Write a program to make use of the **`IsPrime`** function to generate the first 20 prime numbers.

Evidence 5: Your amended program code. **[5]**

Evidence 6: One screenshot showing the output from running the program. **[2]**

3 The librarian of a school library wishes to keep records of the books held in the library. Each book has the following data recorded:

- **CatalogueNo** is used to identify a particular book and is seven digits. The first four digits represent the year that the book was placed in the library and the last three digits are used to make the CatalogueNo unique e.g. 2013103.
- **Title** represents the title of the book and is at most 30 characters.
- **Author** represents the author(s) of the book and is at most 20 characters.
- **Format** is a single character and is used to indicate whether the book is of standard size (S), large print (L), oversize (O) or paperback (P).

A file which contains details of each book has to be created.

Task 3.1

Write a function, **CREATEBOOK**, which when called will allow the user to enter and store data into a text file named **BOOK.DAT**. The file has the following structure.

```
<CreationDate><NumberOfBooks>
<CatalogueNo><Title><Author><Format>
<CatalogueNo><Title><Author><Format>
……..
<CatalogueNo><Title><Author><Format>
```

CreationDate is the date of the file creation and is in the form **DD-MM-YYYY**.
NumberOfBooks is the number of books in the file e.g. 13.

Evidence 7: Your program code for **CREATEBOOK**. **[16]**

Evidence 8: Test data for CatalogueNo explaining the purpose of each test and showing screenshots of the outcomes. **[6]**

Task 3.2

Write a function **DISPLAYBOOK** to read the data from the text file **BOOK.DAT** and display them on the screen clearly for the user in the following format:

```
Date: 16-09-2013                          Number of books: 4

No. Catalogue  Title                      Author                Format
1.  2013001    The Mystery of Pioneers    Steve Bones           L
2.  2013002    The Mega Pioneers          Penny Lim             S
3.  2013003    Missing Pioneers           Yap Jun Yee           O
4.  2013004    Marvellous Pioneers        Henry Ford            P
```

Evidence 9: Your program code for **DISPLAYBOOK**. **[11]**

Evidence 10: Screenshot from running **DISPLAYBOOK**. **[2]**

4 A supermarket keeps a computerised stock control system. The master file which contains details on each product is sequentially organised. The master file contains fields such as ***ProductID, ProductName, QuantityInStock, Price, Weight***. A transaction file which contains fields such as ***ProductID*** and ***QuantityOut*** is used to update the master file every day. The ***QuantityOut*** in the transaction file is used to update the ***QuantityInStock*** in the master file for the corresponding product. For example, if ***QuantityOut*** for product A is 5 in the transaction file, then 5 is subtracted from ***QuantityInStock*** for product A in the master file.

Task 4.1

Write a procedure **SORT_TRANFILE** to sort the transaction file **TRANSACTION.txt** according to the key of the master file, ***ProductID***. Save the sorted file as **S_TRANSACTION.txt**. Ensure that the header (first line) is copied over from **TRANSACTION.txt** to **S_TRANSACTION.txt**.

Evidence 11: Your program code. **[9]**

Task 4.2

The following is an algorithm for generating a new master file using the sorted **S_TRANSACTION.txt** and the existing master file **MASTERFILE.txt**.

```
Open master file for reading
Open transaction file for reading
Open new master file for writing
Repeat
    Read next transaction record
    While master record key < transaction record key
        Write master record to new master file
        Read next master record
    EndWhile
    Update record
Until End of File (transaction)
While Not End of File (master)
    Read next master record
    Write master record to new master file
EndWhile
```

Write a procedure **GEN_NEWMASTER** to generate a new master file **NEWMASTERFILE**.txt using the above algorithm. Ensure that the header (first line) is copied over from **MASTERFILE.txt** to **NEWMASTERFILE**.txt.

Evidence 12: Your program code. **[14]**

Evidence 13: The **NEWMASTERFILE**.txt **after** running your program. **[2]**

End of Paper