IB DIPLOMA PROGRAMME
PROGRAMME DU DIPLÔME DU BI
PROGRAMA DEL DIPLOMA DEL BI

22067014

COMPUTER SCIENCE
STANDARD LEVEL
PAPER 2

Wednesday 3 May 2006 (morning)

1 hour 30 minutes

INSTRUCTIONS TO CANDIDATES

- Do not open this examination paper until instructed to do so.
- Answer all the questions.

1. The string array `names` has been declared as shown below:

```
String[ ] names = new String [100];
```

The global integer variable `entries` stores the current number of names in the array.

At present, the array only contains the following 6 items:

	0	1	2	3	4	5
names	sara	surita	juan	alvaro	nicole	peter

e.g. `names[0] = sara ,  entries = 6`

The method `delete( )`, partially written below, deletes the array item with index `position` and moves all items with index greater than `position`, one place to the left.

e.g. if position = 2, then the data item `juan` would be deleted from the names array, and names[2] would become equal to `alvaro`, names[3] equal to `nicole` and so on.

```
// the data item with index 'position' is deleted from the names array
public void delete(String[ ] names, int position) {

     int i= position + 1;
     while (i < entries) {
            names [i - 1] = names [i];
            i = i + 1;
     }
     entries = entries - 1;
}
```

(a) Explain what the information in the method signature tells us about the method `delete( ).` *[2 marks]*

(b) Explain what is meant by the scope of a variable, by referring to the variable `i` in the method `delete( ).` *[3 marks]*

(c) Complete the trace table for the algorithm given the method call `delete(names, 3).` *[3 marks]*

position	i	names[0]	names[1]	names[2]	names[3]	names[4]	names[5]
3							

(This question continues on the following page)

(Question 1 continued)

(d) Outline how the use of the variable `entries` can prevent the displaying or printing of duplicate names from this array. *[2 marks]*

A mobile phone uses the **two** parallel arrays `names` and `numbers` to store up to 100 entries in its address book.

e.g. the first entry might be stored as:

```
names[0] = 'sara', numbers[0] = '68594753'
```

If the address book is not full, all empty positions are placed at the end of the arrays.

(e) Construct the method `addEntry()` that adds a new entry (name and number) to the address book, if it is not yet full. Assume that the global variable `entries` stores the current number of entries in the address book. *[4 marks]*

When the mobile phone receives an incoming phone call, the method `findName( )` is called, which searches through the address book looking for the telephone number of the caller.
If it is found, the method returns the caller's name. A suitable message is returned if the number is not found.

(f) Construct the method `findName( )`. *[6 marks]*

2. An engineering company is designing a new signaling system for a section of railway line.

The analysis team are using a prototyping approach as part of their design process.

(a) Explain the advantage of using this approach for

(i) the end-users (the railway company). *[2 marks]*

(ii) the systems analysis team. *[2 marks]*

The implementation will include the installation of a large screen in the signalling control centre, which will show a display of the railway network together with the position of all trains and the status of all signals. The centre also updates a similar display in the stationmaster's office of each station. The system includes centrally installed programs that monitor and set the signals on the track.

(b) Describe **one** way in which sensors can capture the position of each train. *[2 marks]*

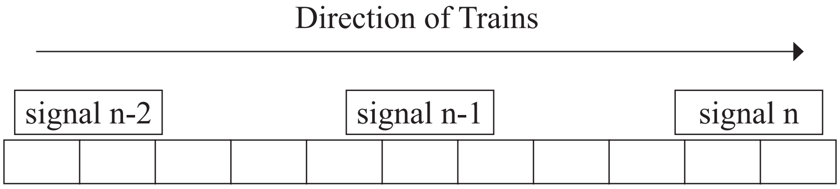
The data taken from the sensors is stored on magnetic tape so that it can be replayed if necessary.

(c) Explain which type of processing is needed

(i) to control the signaling. *[2 marks]*

(ii) for the storage and retrieval of sensor data. *[2 marks]*

Consider the stretch of track shown below. The trains move only in the direction shown and the signals are numbered in order starting from 0.



The algorithm that controls the state of the signals needs two data structures:

One records for every signal, n, whether a train has just passed that signal and not yet passed the next signal. (i.e. is between the first signal and the next).

The other records for every signal the state of the signal: **0** for red, **1** for yellow, and **2** for green.

(This question continues on the following page)

(Question 2 continued)

(d) State **two** suitable data structures. *[2 marks]*

As a train passes signal n, that signal is changed to red, the previous signal (n-1) to yellow and signal (n-2) to green.

(e) Complete the algorithm fragment started below, that monitors whether a train has passed each of the signals, and sets the signals as appropriate. *[4 marks]*

```
// totalSignals represents the number of signals in this
// section of track
for (int n = 0; n < totalSignals; n++) {
         // one or more lines are missing
}
```

(f) Discuss possible back-up strategies that could be implemented which would keep the railway system partially or fully functioning, in the event of a systems failure at the control centre. *[4 marks]*

This question requires the use of the Case Study.

3. (a) Explain the fundamental difference between the way that the MIDI format stores data compared to music files stored on CDs and cassette tapes. *[2 marks]*

On a piano, the note of Middle C is struck and held for 5 seconds after which it is released.

(b) Compare the storage requirements of a MIDI file with that of a music file stored on a CD, referring to the example given (actual storage values are not required). *[3 marks]*

A recently formed band wish to make a promotional recording of their music. However, as they are not yet accomplished musicians they are finding difficulty with the playing of certain pieces of music.

(c) Explain **two** ways in which a computer-based sequencer could help to improve the quality of their recorded music. *[4 marks]*

(d) The most common MIDI channel voice instruction is **NOTE ON,** which consists of 1 status byte followed by 2 data bytes.

The general format of a **NOTE ON** instruction is as follows:

Status Byte	Data Byte #1 (pitch/note)	Data Byte #2 (volume)
1001nnnn	0xxxxxxx	0xxxxxxx

where, (**nnnn + 1**) represents the channel number, and **xxxxxxx** represents a decimal value between 0 and 127.

For example: 10011010 00111100 00000001
represents **NOTE ON** on channel 11 of note #60 (MIDDLE C) at a volume of 1 (the lowest possible volume).

(i) State which channel number is selected by the status byte 10010011. *[1 mark]*

(ii) Describe how the receiving device (e.g. a sound module) distinguishes between a status byte and a data byte. *[2 marks]*

(iii) State the number of different notes that can be played using this format. *[1 mark]*

(iv) Identify the bytes that would correspond to the following instruction:
NOTE ON using Channel 12, playing Middle C at maximum volume. *[2 marks]*

(This question continues on the following page)

(Question 3 continued)

(e) The instruction bytes are often written down in hexadecimal. As an example, the byte 10011010 (binary), could be written as 9A (hex), with each hexadecimal digit replacing a group of 4 binary bits.

(i) State why the instructions are often written in hexadecimal. *[1 mark]*

(ii) The **NOTE ON** example given previously in part (d) was:

10011010 00111100 00000001

Given that the status byte for **NOTE OFF** is represented by 1000nnnn, identify, using hexadecimal notation, the instruction that will turn this note off. *[3 marks]*

(f) Small scale MIDI files are now being commonly found in everyday applications outside of their original use in the recording studio. Outline **two** such applications. *[4 marks]*

(g) Explain **two** features of MIDI sound modules that allow musicians to produce fuller, more varied sounds. *[4 marks]*

(h) Discuss whether singing can be handled effectively using MIDI technology. *[3 marks]*