



**Award Summary**

Outstanding Achievement (OA)	56
High Achievement (HA)	94
Satisfactory Achievement (SA)	106
Reassessed into neighbour	35
Total candidates	291

**Gender Breakdown**

Males	267
Females	24

**Ratings awarded (internally and externally)**

Criterion	A		B		C		D	
	int	ext	int	ext	int	ext	int	ext
Criterion 1	71	112	106	102	92	45	15	12
Criterion 2	120		100		60		3	
Criterion 3	91		84		99		9	
Criterion 4	81		107		89		7	
Criterion 5	72	54	90	95	104	102	18	20
Criterion 6	71	96	90	88	97	47	23	38
Criterion 7	71	82	89	67	111	99	12	23
Criterion 8	109		83		67		25	
Criterion 9	77		79		108		20	
Criterion 10	87	47	95	91	92	108	10	24



### **General Comments**

This is the first year that IF862 Computer Science has been taught with Java as the programming language. The enrolment has increased dramatically since last year and there also appears to have been an overall improvement in the level of attainment.

A feature of the results was that they suggested a clear dichotomy between those who understood Java and those who did not. There were many A ratings in the Java sections. The number of B's was then significantly less. Another large group of candidates were rated C or less in these sections indicating a very basic understanding of programming.

Other comments are specific to individual criteria. They are interspersed with a copy of the examination and answers on the following pages.

In conclusion, there was no adverse feedback about the exam from staff or students

### **Section A – Criterion 1**

#### **General Comments**

Most students handled this section particularly well, partly because they found the questions easy to understand, ie. they were familiar with the scenarios and the style given, and the questions followed each other logically.

In some cases, though, there was a reluctance to use the format given. GOTO statements particularly stood out as inappropriate (often linked with other Basic-type commands).

#### **Question 1**

The following algorithm is designed to be used by a microwave oven. The oven has two buttons – a 'Time' button to add 10 seconds onto the time the food will be cooked for, and a 'Start' button to commence cooking after a time has been entered.

- 1 ***Initially***
- 2 Set time to zero seconds
- 3
- 4 ***When Time button is pressed***
- 5 Set time to time + 10 seconds
- 6 PRINT time
- 7
- 8 ***When Start button is pressed***
- 9 Run oven for time seconds

- 
- (a) For safety reasons, the oven should not allow a time larger than 20 minutes to be entered. At what stage (give the line number of the statement **before** your statement) would you include a statement to prevent illegal times from being entered?
- (b) How would you prevent a time greater than 20 minutes from being entered?
- (c) Presently the program runs the oven for 0 seconds if the user presses the 'Start' button before the 'Time' button. How could you test that a time has been entered before starting the oven? Give the statement and position of the statement in terms of the original line numbers.

#### Solutions

- (a) Prevent illegal times by putting an instruction before line 5, eg. to test for 1 200 seconds (20mins).
- (b) If time < 1200 set time to time + 10 seconds
- (c) Make line 9:  
If time > 0 Run oven for time seconds

#### Comments

It was considered necessary to test for the illegal time at the 'data input' stage, ie. before line 5, rather than when the Start button is about to be activated. It would not necessarily be appropriate to reset the time to zero if the maximum time was exceeded (user friendliness!).

Some students wrote unnecessarily convoluted decision statements at line 9.

#### Question 2

An algorithm is to be designed to calculate the cost of purchasing a ticket to a music concert according to buttons pressed by the user specifying the type of ticket required and the method of payment. Once the user has selected these options, they press another button to print the cost of the ticket.

##### ***Ticket Prices***

*Standing room only* \$25.00

*Regular seats* \$35.00

*Deluxe seats* \$70.00

*Payment options are cash or credit card – a discount of 10% is given if payment is made by credit card.*

The algorithm should have default settings of a standing room only ticket paid for with cash. Below is a template used to design the algorithm to calculate the ticket price.

##### ***Initially***

*Set ticket type to standing room*

*Set payment to cash*

*Set ticket price to \$25.00*

Using the template above as an example, design the algorithm to calculate the ticket price:

- (a) when the 'Standing Room Button' is pressed;
- (b) when the 'Regular Seat Button' is pressed;
- (c) when the 'Deluxe Seat Button' is pressed;
- (d) when the 'Cash Payment Button' is pressed;
- (e) when the 'Credit Card Payment Button' is pressed; and
- (f) when the 'Print Ticket Price Button' is pressed.

#### Solutions

- (a) (Note that the default values cover the 'Standing room' option)  
When 'Standing Room' button is pressed  
Print "Standing room selected"
- (b) When 'Regular Seat' button is pressed  
Set ticket type to Regular seat  
Set ticket price to \$35  
Print "Regular seat selected"
- (c) When 'Deluxe Seat' button is pressed  
Set ticket type to Deluxe seat  
Set ticket price to \$70  
Print "Deluxe seat selected"
- (d) When 'Cash Payment' button is pressed  
Print "Please pay cash as shown on the printed ticket"
- (e) When 'Credit Card Payment' button is selected  
Set ticket price to ticket price\*0.9  
Set payment to card  
Print "Please provide your credit card details when your ticket is printed"
- (f) When 'Print Ticket Price' button is pressed  
Print "Seat is" ticket type  
Print "Payment method is" payment  
Print "Cost of ticket =" ticket price

#### Comments

If the style given in Question 1 was followed, this was a straightforward question. Some students attempted to write it in Java, and most students made limited use of the variables given, perhaps highlighted by the question only asking for the ticket cost to be printed.

Some students added variables unnecessarily, and there was some confusion on the allocation of values to string v numeric variables.

This criterion really needed very clear use of object programming models, ie explicit use of 'When ...' to respond to buttons being pressed. Many students used 'If ...' statements at these times.

Pleasingly, a number of students identified the problem of a purchaser repeatedly pushing the 'Credit card' button, hence obtaining multiple 10% discounts. While very few attempted to cure this problem, they were well rewarded if they did.

Also such a program should ideally reset values after printing the ticket.

---

**Question 3**

You have been employed to write an algorithm for a machine to calculate the cost of sending a parcel via a courier company from Tasmania to various other locations. The user places the parcel on a scale which weighs it, and then presses a button to select the destination of the parcel. There are four possible destinations – Within Tasmania, Mainland Australia, New Zealand, and Other Country. Once a destination is chosen the machine will then print the cost.

You may assume the following postage rates:

*Within Tasmania:* \$1.50 per kg  
*Mainland Australia:* \$2.50 per kg  
*New Zealand:* \$3.95 per kg  
*Other Country:* \$12.45 per kg

Using a similar structure to Question 2, design the algorithm to complete this task.

**Solutions****Initially**

Set weight = 0 Kg  
Set Tas\_rate = \$1.50  
Set Aus\_rate = \$2.50  
Set NZ\_rate = \$3.95  
Set OS\_rate = \$12.95  
Set cost = 0

**When Parcel weighed**

Set weight

**When 'Within Tasmania' button pressed**

Set cost = weight\*Tas\_rate  
Print "Weight =" weight  
Print "Cost =" cost

**When 'Mainland Australia' button pressed**

Set cost = weight\*Aus\_rate  
Print "Weight =" weight  
Print "Cost =" cost

**When 'New Zealand' button pressed**

Set cost = weight\*NZ\_rate  
Print "Weight =" weight  
Print "Cost =" cost

**When 'Other Country' button pressed**

Set cost = weight\*OS\_rate  
Print "Weight =" weight  
Print "Cost =" cost

---

### Comments

While a straightforward extension of the previous question, many students provided a bare-bones answer only, particularly re minimal use of variables. Better answers acknowledged the significance of being able to set values in the Initialisation block, rather than searching through the program when changes are needed.

It was also considered a good idea to reset the weight after the sticker was printed.

Again, how much information was desirable to be printed out? The printout had to be part of the destination selection (**not** with an extra button), and some students did include weight and/or destination in the printout.

### Section B – Criterion 5

#### Question 4

- (a) What value will be returned if the following statements are executed, with ch having an initial value of 'u'?

```
ch=Character.toUpperCase(ch);
if (ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U')
    return true;
else
    return false; (lines 11-15)
```

- (b) What is the final value of the variable number after the following statement is executed if number initially has the value 5 and s.charAt(i) returns the value '7'?

```
number = number * 10 + ((int) s.charAt(i) - (int) '0'); (line 30)
```

- (c) Which of the methods (countVowels, convertToInt or isPalindrome) will be called within the paint method if line 45 is changed as shown below?

```
int whichMethod=2;
```

### Solutions

- (a) *True*

- (b) Number      = 5 \* 10 + 7  
                  = 57

- (c) `convertToInt`

### Comments

- (a) This question was fairly well done.
- (b) A common error was to evaluate s.charAt(i) to 7 and obtain 9 as an answer.
- (c) The question was generally well done.

**Question 5**

- (a) How would the program's behaviour change if line 11 was deleted from the program?
- (b) If line 12 was changed to the following, what effect would that change have on the program?  
if (ch=='A' && ch=='E' && ch=='T' && ch=='O' && ch=='U')
- (c) If whichMethod was initialised to 3 on line 45, what would be displayed on the screen when the program was run?

## Solutions

- (a) Lower case vowels won't be counted therefore the output would be "the answer is 0".
- (c) Since a single character can't contain multiple values, isVowel will always return false, therefore no vowels detected.
- (d) The string is a palidrome.

## Comments

- (a) Many said that Line 12 would always return false – (which it will, but only for the data given: "glenelg") and did not mention that it will still evaluate to true for some uppercase values. Better answers explained that it would only check for uppercase values.
- (b) It was important to explain that ch could only contain one value and hence the expression would always evaluate to false. Many confused ch as a string and stated it would have to contain all the values A E I O U.
- (c) For whichMethod = 3 many were unable to correctly analyse the nested if/else statements in the paint method and decided that the program could not function at all.

**Question 6**

- (a) How many times will the following loop be executed if s has the value "madam"?

```
for (int i=0; i<s.length()/2; i++)  
    if (s.charAt(i)!=s.charAt(s.length()-i-1))  
        palindrome=false;      (lines 37-39)
```

- (b) List the local variables in the following program and indicate where within the class they may be used.
- (c) The programmer wishes to eliminate the use of class variables within the countVowels method. Show how this can be accomplished by using a parameter, local variable and return value.

## Solutions

(a) Loop executes twice ( I = 0, 1)

local variable	used
ch	method isVowel
number	convertToInt
palindrome	isPalindrome
whichMethod	paint
optional* i (in loop)	

(c) 

```
public int countVowels (string s)
{
    int result = 0;
    for (int i = 0; i < s.length(); i++)
    {
        if (isVowel (s.chaAt(i)))
            result++;
    }
    return result;
}
```

call:  
answer = countVowel(myString)

## Comments

- (a) A large number of students incorrectly calculated 3 as the number of loop iterations.
- (b) Many students overlooked ch as a local variable to isVowel and whichMethod as local to the paint method. Another common omission was a failure to mention the scope of use for the variables.
- (c) Those who did this question generally did it well. A common oversight was to provide an integer parameter rather than a string parameter whilst retaining the class variable myString. Good answers also included a modification to the method call at Line 49.

```
1 import java.applet.Applet;
2 import java.awt.*;
3
4 public class SectionBQuestion extends Applet
5 {
6     String myString="glenelg";
7     int answer;
8
9     public boolean isVowel(char ch)
10    {
11        ch=Character.toUpperCase(ch);
12        if (ch=='A' || ch=='E' || ch=='I' || ch=='O' || ch=='U')
13            return true;
14        else
15            return false;
16    }
17
18    public void countVowels()
19    {
20        answer = 0;
21        for (int i=0; i<myString.length(); i++)
22            if (isVowel(myString.charAt(i)))
23                answer++;
24    }
25
26    public int convertToInt(String s)
27    {
28        int number = 0;
29        for (int i=0; i<s.length(); i++)
30            number = number * 10 + ((int) s.charAt(i) - (int) '0');
31        return number;
32    }
33
34    public boolean isPalindrome(String s)
35    {
36        boolean palindrome=true;
37        for (int i=0; i<s.length()/2; i++)
38            if (s.charAt(i)!=s.charAt(s.length()-i-1))
39                palindrome=false;
40        return palindrome;
41    }
42
43    public void paint( Graphics g )
44    {
45        int whichMethod=1;
46        if (whichMethod<3)
47        {
48            if (whichMethod==1)
49                countVowels();
50            else
51                answer=convertToInt(myString);
52            g.drawString("The answer is " + answer, 10, 100);
53        }
54        else
55            if (isPalindrome(myString))
56                g.drawString("The string is a palindrome", 10, 100);
57            else
58                g.drawString("The string is not a palindrome", 10, 100);
59    }
60 }
```

**Section C – Criterion 6****Question 7**

- (a) A 4-bit two's complement system can represent 16 different integer values whilst a 4-bit signed representation can only store 15 different integers. Explain why this is so.
- (b) The UniCode character set uses 16 bits to represent each character. How many different characters can be represented by a 16-bit binary code? Explain how you obtained this value.

**Solutions**

- (a) Two's Complement  $+7 \rightarrow -8$  Only one representation of zero  
Signed magnitude  $+7 \rightarrow -7$  Two representations of zero (+0 and -0)
- (b) Unicode is a 16 bit representation, therefore has  $2^{16}$  characters = 65536.

**Comments**

- (a) Most students answered this question sufficiently well to rate a C. Well explained by better students but poorer students didn't demonstrate an understanding of two's complement.
- (b) Most students arrived at  $2^{16}$  but couldn't always explain why.

**Question 8**

- (a) Convert  $0.4_{10}$  to binary (limit your answer to 7 bits after the decimal point). Show your working.
- (b) (i) Calculate the result of  $0.4_{10} + 0.4_{10}$  using 5 bits to represent each number and the answer, and convert the result to decimal.
- (ii) Repeat this process using a 7-bit representation.
- (c) Explain the difference between the answers found in part (b). What implications does this have for programming?

**Solutions**

- (a) Use multiplication by 2 to convert decimal to binary  $0.4_{10} = 0.0110011_2$
- (b) (i)  $0.01100_2 + 0.01100_2 = 0.11000_2$   
 $= 0.5_{10} + 0.25_{10}$   
 $= 0.75_{10}$
- (ii)  $0.0110011_2 + 0.0110011_2 = 0.1100110_2$   
 $= 0.5_{10} + 0.25 + 0.03125 + 0.015625_{10}$   
 $= 0.796875_{10}$

- (b) From this emphasise and expand on the following points –

the greater the number of bits the greater the accuracy,  
 choice of data type float or double,  
 inexact representation causes may cause infinite looping unless the programmer caters for this.

#### Comments

This question was well answered by most students. A common problem was converting binary back to decimal. Many students rounded the 5 bit representation up so that answers were inaccurate. Some people included the 0 before the binary point as one of the bits.

- (c) These answers were often brief and didn't necessarily draw out the implications for programming.

#### Question 9

- (a) (i) How many hosts can be supported by a Class C address if no subnets are used?  
 (ii) If we divide this address into 5 subnets, how many bits will we need to borrow from the host field to create these subnets?  
 (iii) How many useable hosts can be supported within each subnet?
- (b) If we ignore the different classes of address and the presence of network and broadcast addresses, the maximum number of hosts supported by the current IP addressing scheme is  $2^{32}$ . What is the maximum number of hosts supported if we take these issues into account?

#### Solutions

- (a) (i) Number of hosts supported by Class C address =  $2^8 - 2 = 254$   
 (ii) Number of bits required for 5 subnets = 3  
 (iii) Number of useable hosts per subnet =  $2^5 - 2 = 30$
- (b) Class A = 127 networks x  $(2^{24} - 2)$  hosts = 2 130 706 178  
 Class B =  $(2^6 \times 2^8)$  networks x  $(2^{16} - 2)$  hosts = 1 073 709 056  
 Class C =  $(2^5 \times 2^8 \times 2^8)$  networks x  $(2^8 - 2)$  hosts = 532 676 608  
**Total 3 737 091 842**

#### Comments

- (a) This was generally well answered.  
 (b) This was only successfully completed by students who clearly understood IP addressing and classes.

**Section D – Criterion 7****Question 10**

Consider the code for an Applet given below. Draw a rough sketch of what you would expect the output from the Applet to look like on the screen. Be sure that you clearly distinguish between buttons, labels and textfields.

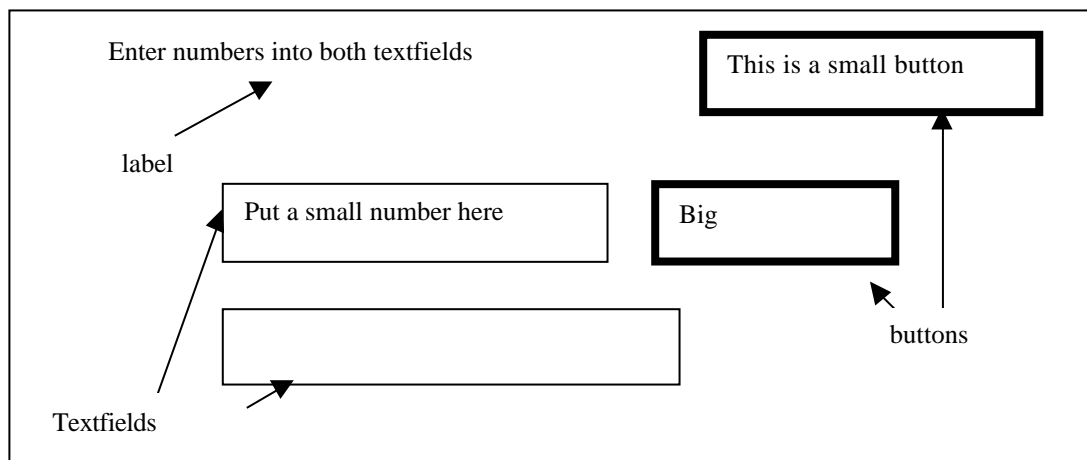
```
import java.applet.*;
import java.awt.*;

public class ExamQuestion10 extends Applet
{
    Label prompt;
    Button smallButton, bigButton;
    TextField smallText, bigText;

    public void init()
    {
        prompt = new Label("Enter numbers into both textfields");
        smallButton= new Button("This is a small button");
        bigButton=new Button("Big");
        smallText = new TextField ("Put a small number in here");
        bigText= new TextField(30);
        add(prompt);
        add(smallButton);
        add(smallText);
        add(bigButton);
        add(bigText);
    }

    public void paint(Graphics g)
    {
    }
}
```

Solutions



**Comments**

Most students successfully completed this question however, common errors were the first text field was often missed and the two buttons were often shown before the text fields.

**Question 11**

The following class has been defined in Java:

```
public class Product
{
    String name;
    int priceInCents;

    Product(String thisName, int price)
    {
        name = thisName;
        priceInCents = price;
    }

    public String getName()
    {
        return name;
    }

    public int getPrice()
    {
        return priceInCents;
    }
}
```

- (a) A student wishes to create Product objects in an Applet. The required objects are:

product1 with a name of "Snow Crash" and a price of \$12.95.  
product2 with a name of "Leftism" and a price of \$30.95.

Show how these objects could be declared and created.

- (b) The student would also like to see if the objects have been correctly created so decides to print the name and price of each Product onto the screen. How could this be accomplished from the Applet?

## Solutions

(a) Declaration:

```
Product product1, product2;
```

Creating (initialising) in init()

```
product1 = new Product("Snow Crash", 1295);
product2 = new Product("Leftism", 3095);
```

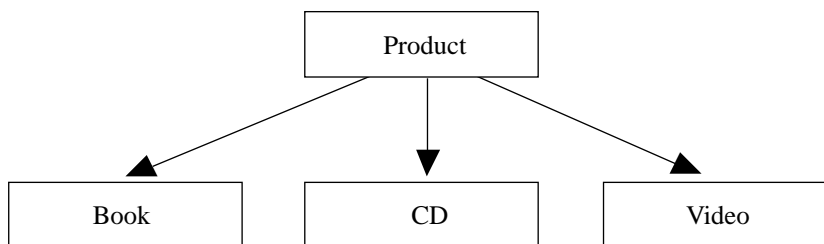
(b) Within paint method:

```
g.drawString("The first product is " + product1.getname() + "with a price $" + product1.getprice(),
20, 20);
```

```
g.drawString("The second product is " + product2.getname() + "with a price $" + product2.getprice(),
20, 40);
```

**Question 12**

The student decided to create some further objects as defined in the hierarchy below.



```

class Book extends Product
{
    int numberOfPages;

    public Book(String thisName, int price)
    {
        super(thisName, price);
        numberOfPages = 0;
    }

    public void setPages (int pages)
    {
        numberOfPages=pages;
    }

    public int getPages ()
    {
        return numberOfPages;
    }
}
  
```

- 
- (a) Show how myBook (a Book) could be created with a name of "Neuromancer" and a price of \$10.50.
- (b) Show how you would set the number of pages to 254.

#### Solutions

- (a) Declaration:

```
Book myBook;
```

Initialise:

```
MyBook = new Book ("Neuromancer", 1050);
```

- (b) myBook.setPages(254)

#### Comments

#### Q11 and 12

Mainly students either knew this section or didn't. The main problem was many didn't understand the use of constructors. A large number of students did not understand the use of accessor methods either. Because of the structure of the class variables, they could get away with this lack of knowledge and directly reference the variables rather than use the get... and set... methods.

There was some confusion between variable names and values stored in variables including some instances of literals on the left hand side of equivalence statements

As always, many didn't carefully read the questions misspelling given methods and not using given variable names.

### Section E – Criterion 10

#### General Comments

This section was reasonably well done, however some candidates wrongly believed that they could correctly answer questions by copying sections from their notes and not applying them to the questions. For example some candidates listed cables and their features without applying their reasoning to the scenario provided. It was pleasing to note that no candidates did both parts in the questions involving choices.

#### Question 13

- (a) A school consisting of two separate buildings wants to install a computer network. The buildings are separated by a distance of 150 m and lie on either side of a public road. The network is to have a minimum of 50 network connections. What type of media would you suggest for network connections between buildings and within the buildings? Give reasons for your answers.

As a guide your answer to this question should be no longer than half a page.

- (b) Consider the IP address 93.255.255.255
- What class of IP address is this?
  - What is the significance of the final 3 bytes of the address?
- (c) Draw a combinational circuit to implement the following sentence of the propositional logic:

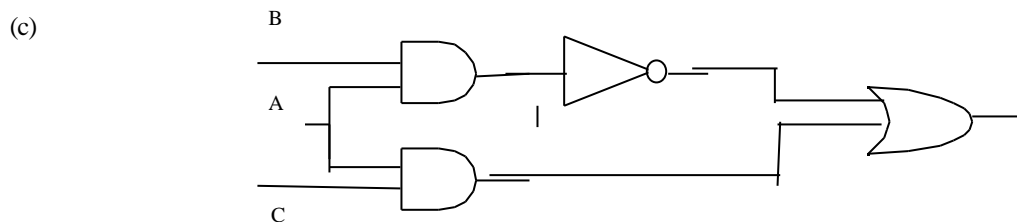
$$\sim (A \vee B) \vee (A \wedge C)$$

### Solutions

- (a) Since the buildings are separated by a public road we will need to use a carrier such as Telstra to provide the link between the buildings if we wish to use cable. This may involve sending a signal over either twisted pair or optical fibre. Another option would be to link the buildings using microwave technology. This technology requires line of sight and may be cheaper than paying regular carrier charges. Speeds of 2Mbps are possible over 30km. Thus I would recommend a microwave link.

Within the building there are many possibilities. The most cost effective is UTP providing speeds of 1Gbps with a maximum distance of 100m. The cable is subject to EMI but this is unlikely to be a problem within a building. Other options that could be considered are fibre to the desktop, coaxial cable, STP or a wireless network. Coaxial cable is old technology and likely to be phased out of the standard and hence not a good choice. Shielded Twisted Pair (STP) has the same arrangement as UTP but has a shield around the cables, which is grounded at each end. This cable is much more difficult to terminate and hence probably not a good choice (more expensive.) Optical fibre could be considered depending on finances. It can have speeds over 1Gbps, is noise resistance and cover a distance of up to 3000m. The cost of the cable, installation and network cards may rule out this option. Hence I would recommend UTP cable.

- (b) (i) Class A address
- (ii) This is the broadcast address for the class A network. (Note some students indicated that the last 3 bytes identified the host. This answer was accepted.)



### Comments

Many candidates had problems with the order of operation for example sketching  $\sim A \vee \sim B$  instead of  $\sim(A \vee B)$ .

**Question 14**

Answer part (a) OR part (b) of this question.

- (a) (i) How many unique MAC (Media Access Control) addresses are available in the Ethernet protocol? Express your answer as a power of 2.
- (ii) Each MAC address is divided into two sections. What aspect of the interpretation of these sections makes it very unlikely that all possible addresses will ever be used?

OR

- (b) Trace the contents of the accumulator during the execution of the following SAM program:

**Program**

<i>Address</i>	<i>Operation</i>	<i>Operand</i>
	LD	NUMBER
	LS	15
	RS	15
	BZ	SKIP
	LD	LETTERO
	WR	
	B	END
SKIP	LD	LETTERE
	WR	
END	H	

**Data**

<i>Address</i>	<i>Contents</i>
NUMBER	0000 0000 1001 0110
LETTERO	0000 0000 0100 1111
LETTERE	0000 0000 0100 0101

- (i) What character will be output when this program is run using the data shown above?
- (ii) The program produces two possible output characters. What is the relationship between the value of NUMBER, and which character is displayed by the program?

Solutions

**Part A**

- (i) A MAC address occupies 48 bits, therefore there are  $2^{48} = 2.81 * 10^{14}$  addresses.

- (ii) MAC addresses are 48 bits in length and are expressed as twelve hexadecimal digits. The first six hexadecimal digits, which are administered by the IEEE, identify the manufacturer or vendor and thus comprise the. The remaining six hexadecimal digits comprise the interface serial number, or another value administered by the specific vendor.

This would mean that each vendor would have to exceed the production of  $2^{24}$  network cards, which is 16777217 cards. In addition it would be possible to have 16777217 different vendors. Although it may be necessary to allocate a vendor more than one manufacturing card it is unlikely that all possible addresses would be used.

### Part B

Instruction	Acc	Output
LD Number	0000 0000 1001 0110	
LS 15	0000 0000 0000 0000	
RS 15	0000 0000 0000 0000	
BZ skip		
LD Lettere	0000 0000 0100 0101	
Wr		'E'

- (i) The letter E will be printed.
- (ii) If the number is even E is printed otherwise O is printed.

### Comments

Many candidates did not trace the SAM program given in question 14. The placement of the instruction to trace may have been better included with the question in part (i). The LS 15 and RS 15 posed a problem with many students ending up with the same number they began with rather than 0000000000000000. The BZ posed a problem for some students as they repeated the whole program rather than realising the significance of the branch. Part (ii) was poorly done with students not realising the significance of the end digit.

Many students mistook MAC addresses for IP addresses giving the answer  $2^{32}$ .

### Question 15

**Answer part (a) OR part (b) of this question.**

As a guide your answer to this question should be no longer than half a page.

- (a) The OSI model is divided into seven layers. Explain how information flows through these layers when a message is sent from one host to another via a direct link. Discuss the benefits of dividing networking protocols into layers.

**OR**

- (b) Whenever multiple hosts have access to the same network medium, there exists the possibility of a collision occurring (two hosts transmitting at the same time). Discuss **two** different approaches which have been taken in the Data Link Layer to address the problem of collisions.

Solutions

### Part A

If one computer wants to send data to another computer, the *data* must first be packaged by a process called *encapsulation*. Then, as the data moves down through the layers of the OSI model, it receives headers, footers, and other information.

The information starts at the application layer with the appropriate protocol eg http, pop3 etc. The *application layer* identifies and establishes the availability of intended communication partners, and synchronizes cooperating applications. The *presentation layer* will ensure the data is in the appropriate format for transmission. The *session layer* coordinates applications as they interact on two communicating hosts. The *transport layer* may segment the data and add a header specifying the segment number etc. It may also specify window size and acknowledgements. The *network layer* will have a header with identifying source and destination IP addresses. The *data link layer* will add source and destination MAC addresses, CRC etc. The *physical layer* sends the bits across the link. On arrival the data flows back up the layers where each layer is responsible for removing the headers and trailers added in the source.

The OSI model provides a framework for vendors. It standardizes network components to allow multiple-vendor development and support thus allowing different types of network hardware and software to communicate with each other.

It prevents changes in one layer from affecting the other layers, so that they can develop more quickly. It breaks network communication into smaller parts to make learning it easier to understand.

### Part B

Two different techniques for controlling collisions include token passing and CSMA/CD. Ethernet uses CSMA/CD for media access. A station listens for traffic on the segment. If there is no traffic it will begin the transmission. If two transmissions occur simultaneously a collision occurs. A transmitting station continues to listen and hence a collision will be detected and a jamming signal sent. All stations will then cease transmitting for a specific time determined by the backoff algorithm. (different for different machines). The more devices on a segment the more collisions there will be. No station has priority for transmitting.

A node can only transmit a single frame when it receives a token. (token passing). Stations are given turns at receiving tokens (priorities can be assigned). If a station has nothing to transmit the token is passed to the next node. If it has a frame to transmit it attaches the frame to the token. A frame circulates the ring until it reaches the destination where a copy is made. The destination station tags the frame as copied and the frame continues until it reaches the source node where it is removed. Only one token can be circulating at any one time. A token ring network is known as a deterministic network because no station can dominate the network as in a CSMA/CD network such as Ethernet.

( A discussion of switching was also accepted as an answer.)

Comments

Many students described the OSI model in question 15 without addressing the information flow, or discussing benefits. In part b many candidates did not understand the fact that they needed to address collisions and the Data Link layer.

**Some notable answers included:**

A hellofalot addresses

I don't know I can't find my stuff.

The advantage of using these 7 layers is that the data is wrapped in so many layers that it will be quite safe from corruption and contamination.

Many students worried about road noise and the noise made by students.

**All correspondence should be addressed to:**

Tasmanian Secondary Assessment Board

PO Box 147, Sandy Bay 7006

Ph: (03) 6233 6364 Fax: (03) 6224 0175

Email: [reception@tassab.tased.edu.au](mailto:reception@tassab.tased.edu.au)

Internet: <http://www.tassab.tased.edu.au>