COMPUTER SCIENCE (ITC315118)

FEEDBACK FOR STUDENTS AND TEACHERS

MARKING COORDINATOR COMMENT

The markers were happy with this year's exam papers and were pleased to award many 'A' ratings and few 't' ratings. A slight change in the paper (multiple choice questions in the C rating sections) enabled students to display more knowledge and helped the markers to allocate fewer 't' awards.

Section C (Criterion 3) once again caused some difficulties with many students unable to attempt questions 8 and 9 whilst others, with a better understanding of Objects, had little difficulty getting an A rating. **It is important to attempt all questions** and not just rely on getting all the C rating question correct. In Section C many students were unable to answer enough questions correctly for a C rating in question 7 and then wrote nothing in the other two parts.

SOME GENERAL COMMENTS

Read the questions carefully.

When writing algorithms or JAVA code, use indenting conventions. Explanations would also be useful when writing TOY code.

For any questions that involve calculations, show your working. You should do this even if the question doesn't explicitly request it. It is especially important to show your working in the 'A' standard questions of Sections B, D and E. It is difficult to award part marks for an incorrect answer without some explanation of how you obtained that answer.

Make sure you are familiar with the information booklet so that you can look up any JAVA code that you are not familiar with quickly.

SECTION A

QUESTION 1

Initially 1

```
set media = 'V'
set member = 'N'
set days = 5
set rate = 1
set period = 5
```

When a letter is entered into the "Member" Textfield

```
set member to value in "Member" TextField
if member is not equal to 'N' or 'M' or 'E'
      set member = 'N'
if member equals 'N'
      set rate = 1
if member equals 'M'
      set rate = 2
if member equals 'E'
      set rate = 3
display "Membership is: " member
```

When a letter is entered into the "Media" Textfield

```
set media to value in "Media"  TextField
if media is not equals to 'T' or 'V'
      set media = 'V'
if media equals 'T'
      set days = 30
if media equals 'V'
      set days = 5
display "Media is: " media
```

When the "Calculate" button is pressed

```
set period = days * rate
display "Access period is: " period "days."
```

When the "Bonus" button is pressed

```
set period = period + 5
display "Access period is: " period "days."
```

a.(i) The initial value of the variable rate is (1) 5 N M

a.(ii) The number of text fields is 1 (2) 3 4

a.(iii) Not valid value for member E M N (T)

b.(i) The value of rate for an executive member is **3**.

b.(ii) The period for an executive member to access a video is **15 days**.

c.(i) If media is not equal to text or video make media equal to video. (see code)

c.(ii) Add a bonus button to add 5 to the period each time it is pressed. (see code)

EXAMINER COMMENT

Question 1 was done well by most students. Part (b) caused the most trouble, with some students losing marks for not clearly or correctly indicating the position of the required modifications. A number of students attempted to add an else-statement after line 23, but most of these neglected to replace the "if" in line 22 with an "else-if".

QUESTION 2

Initially

```
set drinks = 1
set burgers = 1
set fries = 1
set total_cost = 11
```

When a number is entered into the "Burgers" Textfield

set burgers to value in "*Burgers*" TextField

When a number is entered into the "Fries" Textfield

set fries to value in "*Fries*" TextField

When a number is entered into the "Drinks" Textfield

set drinks to value in "*Drinks*" TextField

When the "Calculate" button is pressed

```
set totalCost = 0.00
while burgers >= 1 and  fries >= 1 and  drinks >= 1
      set totalCost = totalCost + 11.00
      set drinks = drinks - 1
      set burgers = burgers - 1
      set fries = fries – 1
while burgers >= 1 and fries >= 1
      set totalCost = totalCost + 9.00
      set burgers = burgers - 1
      set fries = fries - 1
set total_cost = total_cost + burgers * 8.00
set total_cost = total_cost + fries * 2.50
set total_cost = total_cost + drinks * 3.00
display "Total cost is: " total_cost
```

(a) What is the cost if you buy two burgers, two fries and two drinks? **Cost = $22.00**

(b) Add cost of extra fries and drinks (see code)

(c) Add code for "half meal" (see code)

EXAMINER COMMENT

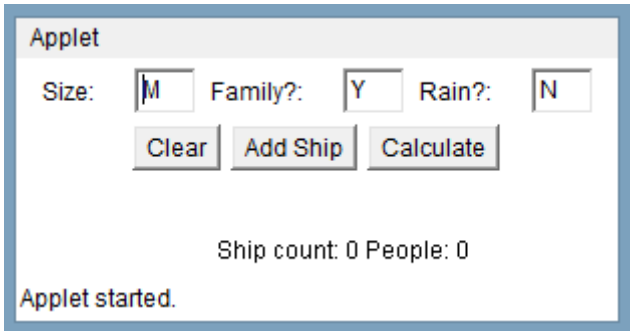
Question 2 was done reasonably well by most students, but many students lost marks for not clearly or correctly indicating the position of the required modifications.

A number of students answered part (b) with overly-complicated structures, such as redundant if-statements. For example…

```
if burgers >=1
      total_price = total_price + burgers * 8.00
```

QUESTION 3

(a)



(b)

Initially

```
set people = 0
set ship_count = 0
set size = 'M'
set family = 'Y'
set rain = 'N'
```

When the "Clear" button is pressed

```
set people = 0
set ship_count = 0
```

When the "Final Calculate" button is pressed

```
set rain to value in "Rain"  TextField
if rain equals 'Y'
        if ship_count equals 1
                set people = people * 0.40
        if ship_count equals 2
                set people = people * 0.50
                if ship_count equals 3
                set people = people * 0.60
display "Extra people would be" people
```

When the "Add Ship" button is pressed

```
set ship_count = ship_count + 1
if ship_count <= 3
    set size to value in "Size" TextField
    set family to value in "Family" TextField
    if size equals 'S'
        set ship = 10
    else
        if size equals 'M'
            set ship = 15
        else
            set ship = 25
    if family equals 'Y'
        set ship = ship * 3
    set people = people + ship
```

EXAMINER COMMENT

Most students attempted Question 3 and were able to demonstrate some understanding of what was required. Students that answered this question satisfactorily demonstrated a wide range of approaches. Some solutions had text fields to enter the size of each of three ships, whereas others had a text field for the number of ships in each category (e.g. small non-family friendly, small family friendly, etc.). A few used the approach shown in the sample solution above. Many solutions made good use of dropdown lists, radio buttons, and/or check boxes.

Many students lost marks in part (a) for not clearly identifying the buttons, text fields, etc. on their screen design.

Marks were often lost in part (b) for providing solutions that depended on the order the buttons or text fields, etc. were used.

Students should be advised that if they are repeating a structure to, for example, set a series of variables to the values from a corresponding series of text fields, it is not always necessary to write out the algorithm in full. As long as consistent variable names and text fields have been used then it may be enough to write out the first structure and then note how is to be repeated.

SECTION B

QUESTION 4

a.(i) Select one: 3 5 (6) 9

a.(ii) Select one: 2 5 (7) 10

a.(iii) Select one: 4 5 8 (16)

b.(i) d = 23

b.(ii) e = 7

b.(iii) f = 'a'

c.(i) h = 6.0

c.(ii) k = 7

k	i
4	
	9
9	
	8
8	
	7
7	
	6

c.(iii) 12.0

m
3.0
6.0
9.0
12.0

d.(i) n = 5

d.(ii) n2 = 3

i	n	N2
	0	0
1		
	1	
		1
2		
	3	
		2
3		
	5	
		3

EXAMINER COMMENT

4.(a) The multiple choice questions were generally well answered, with only the third causing some difficulty with students not knowing what the Math.pow(4,2) function does.

4.b.(i) Many students did not know order of operations correctly and worked left to right.

4.b.(ii) This question was quite well answered, but many students gave incorrect answers of 7.5 (not understanding limitations of ints) or 8 (rounding up to make it an int)

4.b.(iii) This question was general well done, but a surprising number of students gave 'temp' as the answer, given the last line of the code snippet. This shows poor understanding of how variable are assigned in Java.

4.c.(i) This was very well done.

4.c.(ii) This question was very poorly done by about half the students, with "6" being the most common incorrect answer. This "out by one" error is caused by students expecting the "k = 1," line when i is 6, even though the loop has exited by then.

4.c.(iii) This question was generally well answered, but many students gave 9.0 as an incorrect answer, stopping before m got above 10, but not executing the "m=m+3.0" line a final time. A surprising number of students (about 3%) traced the code correctly but made a numerical error stating that 9 + 3 = 11.

4.(d) This question was very well answered, but a common error was to iterate the 'for' loop one too many times.

QUESTION 5

(a) p = 4

P
23
16
8
4

(b) x

1	2	3	4	1
2	3	4	1	2
3	4	1	2	3
4	1	2	3	4

(c) y

0	1	2
9	8	5

i	t	r	y		
			0	1	2
	0	0	5	8	9
0					
	8				
				5	
			8		
1					
	9				
					5
				9	
		1			
0					
	9				
				8	
			9		

EXAMINER COMMENT

5.(a) This question was quite well done, though many students could not determine when p%2==0 correctly, resulting in incorrect operations being done. As long as the loop terminated correctly or int casting was done correctly, partial marks were still given.

5.(b) This question was skipped by a surprising number of students. 2D is a difficult concept for some but the number who made no attempt was still disappointing. Other students turned the given blank 2D array into a trace table for variables i, j and q, which earned no credit. Many students did make a good attempt at the question and any demonstration of 2D arrays garnered some credit. A common mistake amongst these students was to assume that int q is reset to 4 each row. Another common mistake was to incorrectly interpret nested for loops, and only enter values into [0][0] and [1][1] and [2][2] and [3][3]

5.(c) This question confused a lot of students with the condition of the while loop being a method call. Most did not realise that by simply looking at the condition of the while loop, it's only going to exit when the first value is larger than the second which is larger than the third, so the final contents must be 9, 8, 5. Other students misinterpreted the line y[i + 1] = y[i] and added one to the value stored in y[i], turning the 5 into a 6 in the first iteration. Of those who correctly traced the sorting loop, many ran the loop too many times due to misinterpreting the (admittedly complex) end conditions of the loop. This still output the correct final values, and only garnered a slight mark penalty.

QUESTION 6

a.(i) position:

0	1	2	3	4	5	6	7	8	9
2	4	9	13	18	20	21	23	23	25

symbol:

0	1	2	3	4	5	6	7	8	9
'('	')'	'X'	'X'	'□'	'='	'X'	'X'	'='	'□'

a.(ii) *setSymbol*(1) would return '.'

a.(iii) *setSymbol*(2) would return '('

a.(iv) The contents of the array ***pattern*** would be **one** of these two, depending on interpretation:

.	(	.	)	.
.	.	□	.	.
.	=	□	=	.
.	X	.	X	.
X	.	.	.	X

.	.	.	.	X
(	.	=	X	.
.	□	□	.	.
)	.	=	X	.
.	.	.	.	X

b.(i) The scope of variable *temp* is local to body of the method *setSymbol*.

b.(ii) The global variables used in setSymbol are *index, position[]* and *symbol[]*

b.(iii) *position[]* and *symbol[]* could be entered as parameters for the method, since they are not changed, and if they were changed then arrays are reference objects so it would work anyway. However, *index* cannot be passed as a parameter, as it is modified inside the method body, which will not work since *index* is a primitive type and is passed by value.

EXAMINER COMMENT

6.a.(i) The first array was relatively and every student who attempted it got it right, although many students skipped the entire Question 6 and missed out on some easy credit. The second array was generally done well, although some students struggled to interpret the instantiation string, as it had so many ' and '

6.a.(ii) Many students got this right and were able to trace through the method setSymbol with a parameter of 0 and a count of 0. The most common incorrect answer was '(', which would come from assuming the method returns the symbol stored in symbol[0].

6.a.(iii) This was answered correctly by less than half the students, although partial marks were given for any symbol that was stored in the symbol array. 'X' was a very commonly suggested output. Any student who incorrectly determined the symbol array in part ai) but correctly stated here that it was whatever their incorrect first symbol still earned full credit.

6.a.(iv) This question was obviously very challenging, with about 8% of students statewide getting full credit. Anything that showed a mix of '.' and other symbols from the symbol array garnered partial marks, as did anyone who demonstrated some understanding of the spiral pattern, or of the parallel nature of the two arrays position[] and symbol[].

6.b.(i) Many students did not understand the term **scope** and listed possible values it could have been, or some other thing relating to the temp variable. Some students stated the scope was "local" but without stating specifically that it was local to the setSymbol method, only partial credit was gained.

6.b.(ii) This question was generally well answered, with most attempts correctly identifying *index*. Many also identified the two arrays, but many did not consider these global variables. Some also incorrectly included *count* or *temp* in the list

6.c.(iii) This question was very poorly answered by nearly everyone, mostly due to the question being incorrectly interpreted. Many students thought it meant removing the global variables and **instead** making them parameters, which clearly wouldn't work because there's nothing to track their changes. Others interpreted it as a question about **scope,** and suggested that it would work just fine because the local variables would be tracked independently of the global ones. These got some credit, but missed the key point that **index** could not work in this way, because it is modified in the method and this modification would not get passed back to the global variable if it was a parameter of the method.

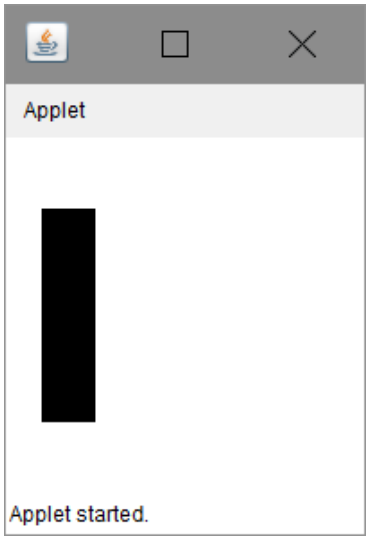
SECTION C

QUESTION 7

a.(i) The screen display will be: "text" (text) texttext "text",2,10

a.(ii) The name of the variable is: Button (open) new Enter

a.(iii)



b.(i) text1 → Then text2 → Then

b.(ii) text1 → Tea text2 → Then

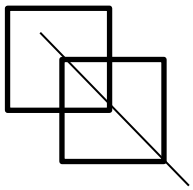
b.(iii) text1 → Back text2 → Then

c.(i) Now!

c.(ii) away

c.(iii) THIS IS NOT RIGHT

(d)



EXAMINER COMMENT

In general this section was well answered although Part© resulted in many incorrect answers. I would stress that it is important to consult the information booklet if you are unsure how some of the string functions operate. Incorrect application of the *substring* method caused most errors. Greater familiarity with the string functions would help in this section.

A startling number of students did not know (or failed to check) that the last two parameters of the drawRect() method indicate the *w*idth and *h*eight of the sides of the rectangle. Students also need to take time setting up a coordinate system to ensure their drawings are correctly scaled and they need to read the question carefully (3 rectangles was a common error!).

QUESTION 8

(a)
```
Weather     city1 = new Weather("Hobart", 13, 25, 20, 3.5);
Weather     city2 = new Weather("Launceston", 9, 28, 10, 5.5);
```

(b)
```
city1.setMaxTemp(23);
city2.setMaxTemp(27);
```

(c)
```
g.drawString(city1.getTown()+ "  Range: "+city1.range(), 100, 100);
 g.drawString(city2. getTown()+ "  Range: "+city2.range(), 100, 200);
```

EXAMINER COMMENT

The students who attempted this question generally had little difficulty. Too many students made little or no attempt on a predictable question and one that is fundamental to the understanding of objects in JAVA. This resulted in many "t" awards as many students were unable to answer sufficient parts of question 7 and needed to show some knowledge in this question. If you are asked to display some output (part (III)) make sure you write correct JAVA code rather than just calling the required method.

QUESTION 9

```
public class Temperature
{
   public   int      max, min, current;
   public   double   humidity, windSpeed, apparent;

   public Temperature(int newCurrent, double newHumidity,
                                          double newWindSpeed)
   {
      current = newCurrent;
      humidity = newHumidity;
      windSpeed = newWindSpeed;
      min = current;
      max = current;
   }
   public double setApparent()
   {
      apparent = current + 0.354 * humidity * current / 100 - 0.70
                                                  * windSpeed - 4.25 ;
      return apparent;
   }
   public void setCurrent(int temp)
   {
      current = temp;
      if (current < min)
         min = current;
      if (current > max)
         max = current;
   }
}
```

EXAMINER COMMENT

Once again this question was well answered by those who attempted it resulting in a high percentage of "A" awards. I would recommend reading the question carefully as some students lost marks by not following exactly what was asked. Many did more than was required.

SECTION D

EXAMINER COMMENT

This section was generally well answered by a large number of students. All of the questions allowed students to show what they knew. It would be good if there was a consistency to the way in which students set out some answers. For example, when drawing a logic circuit it would be preferable to see the circuit run from left to right and lines draw horizontally and vertically. Some diagrams that were drawn bottom to top or right to left. The use of diagonal and very wavy lines made some answers hard to follow. Furthermore, when writing Toy programs it is very useful (maybe essential) to include both the Pseudocode and the explanation column. Some programs were very hard to follow. The explanation is a bonus if you have made a mistake with your contents.

QUESTION 10

a.(i)

A	B	A OR B
F	F	F
F	T	T
T	F	T
T	T	T

a.(ii) *~(~A)* *answer is:* A T ~A F

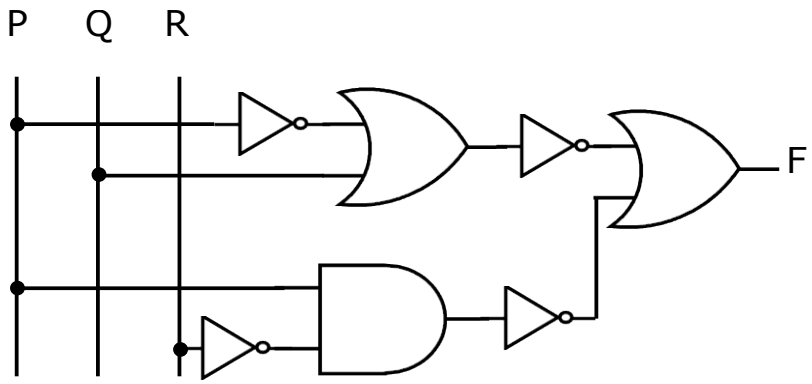
a.(iii) C ≡ ~A ∨ B

b.(i)

A	B	C	~A	~C	~A ∧ B	A ∧ ~C	D
0	0	0	1	1	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	1	1	0	1
0	1	1	1	0	1	0	1
1	0	0	0	1	0	1	1
1	0	1	0	0	0	0	0
1	1	0	0	1	0	1	1
1	1	1	0	0	0	0	0

b.(ii) $E \equiv (\sim A \wedge \sim B) \vee (A \wedge B)$

c.(i)



c.(ii) $G \equiv ((P \vee Q) \wedge \sim P) \wedge \sim(Q \vee R)$

d.(i) The bus used to transfer data between components in the computer. For example transfer data between the memory and the CPU.

d.(ii) Final value stored in Register A is 7.

EXAMINER COMMENT

Part a and b were generally well answered with most students gaining maximum marks. Please note that many students wrote (A ^ B) v ~(A^ B) which is incorrect.

In part c it would be good if all candidates used the setting out of digital logic circuits as shown in part (ii). Some very unusual circuit designs.

Part d was answered correctly by almost all candidates.

QUESTION 11

a.(i) Karnaugh Map for H

$H \equiv (\sim C \wedge \sim D) \vee (\sim A \wedge \sim B) \vee (\sim A \wedge \sim C) \vee (\sim D \wedge \sim B) \vee (A \wedge B \wedge C \wedge D)$

a.(ii) $K \equiv \sim (\sim P \wedge \sim Q) \vee (\sim R \vee \sim R)$

$\equiv \sim\sim(P \vee Q) \vee (\sim R \vee \sim R)$ [L11]

$\equiv (P \vee Q) \vee (\sim R \vee \sim R)$ [L6]

$\equiv (P \vee Q) \vee \sim R$ [L13]

b.(i) The reason the space for the memory address is twice the space for a register is that there are only 16 registers and so only require one byte for the address. There are 256 memory locations and this will require 2 bytes to address.

b.(ii) Before the von Neumann Architecture was invented programs were hard wired into the computer. This meant that changing a program required the program to change the wiring of the computer.

With the von Neumann Architecture the program is stored as a set of instructions in a memory unit that was added to the computer architecture. The advantage of this was that it was easier to change the code by changing the instructions in memory rather than changing the wiring of the computer.

EXAMINER COMMENT

It would be advisable for candidates to show working when simplifying Karnaugh maps. Some credit was given for doing this. Trying to interpret logic expressions is easier when loopings are shown. Many candidates did not apply de Morgans Law when simplifying the equation in (a) (ii). However, part (a) was generally well answered. Part (b) was not attempted by quite a few candidates. The main point to (b) (i) of this question was the difference in the number of locations. It is true that there are double the number of bits when comparing register addresses to memory addresses but there is a difference of 16 to 256 storage locations, which was not highlighted by many candidates. Part (b) (ii) was answered quite poorly with most students failing to mention the change from hard wiring code compared to having code stored as instructions in memory.

QUESTION 12

a.(i)

Memory Address	Contents	Pseudocode	Explanation
01	0001	data	Used for variable x
02	0000	data	Used for variable y
10	8101	R[1] ← mem[01]	Set register 1 to value of x
11	7303	R[3] ← 03	Set register 3 to the value 03
12	7505	R[5] ← 05	Set register 5 to the value 05
13	D116	if (R[1] > 0) pc ←16	If register 1 greater than 0 go to 16
14	C116	if (R[1] == 0) pc ←16	If register 1 equals 0 go to 16
15	C01A	if (R[0] == 0) pc ←1A	Go to 1A
16	2431	R[4] ← R[3] – R[1]	Set register 4 to register 3 – register 1
17	D419	if (R[4] > 0) pc ← 19	If register 4 greater than 0 go to 19
18	C01A	if (R[0] == 0) pc ←1A	Go to 1A
19	9202	mem[02] ← R[5]	Store register 5 into location 02
1A	0000	halt	

a.(ii)

x	y	R[1]	R[3]	R[5]	R[4]
1			3	5	
		1			
					2
	5				

(b) 2412 – This instruction uses the ALU to subtract the contents of register 2 from register 1.

D417 – This instruction uses the ALU to determine if the contents of register 4 is greater than 0.

5113 – This instruction uses the ALU to shift the contents of register 1 by one place to the left.

C013 – This instruction uses the ALU to determine if the contents of register 0 is equal to 0.

EXAMINER COMMENT

This question was either quite well answered or left blank. It would be good practice for candidates to include pseudocode and explanation for each line of Toy code. This will help to make your programs easier to follow. The command C0xx (basically a goto) can be useful to minimise the number of halts you include. In part (b), the ALU "executes the computer's commands involving arithmetic operations and logical comparisons." As such lines 13, 14, 15 and 16 are the lines of code that needed to be commented on. The candidates who attempted this question gave very good answers. However, many candidates wrote nothing.

SECTION E

QUESTION 13

a.(i) $D_{16} = 1101_2$

a.(ii) $1110_2 = 14_{10}$

a.(iii) *ACII code for Character '#' is 35*

(b)

	1	0	1	1	**0**
+	**1**	0	**1**	1	1
1	0	1	1	**0**	1

c.(i) -1 is represented as 11111111.

c.(ii) Adding of numbers create an overflow if the sum of two values exceed the max int value for that representation.

Add 125 + 4

01111101 + 00000100 = 1000001

This calculation has caused an integer overflow so that the result of 129 is actually the representation of –127 because of the 1 in the most significant bit.

d.(i) The 'A' code is 00100 0001 and the 'a' code is 00110 0001.
If the least significant bit is bit 0 then it is bit 5 that can be used to identify the case of the letter.

d.(ii) A programmer is developing a system where 6 bits will be used to represent colours. How many colours will be available?
Since it is a colour it will be an unsigned integer codes from 000000 to 111111 = 2^6 = 64 possible codes.

EXAMINER COMMENT

(a) Surprisingly, a small number of candidates made errors on this question.

(b) Well answered.

c.(i) Well answered, but some gave 11111110 as their answer for -1.

c.(ii) Many candidates didn't explain what overflow is, or work through the suggested example.

d.(i) This question asked which bit determined upper/lower case. Many students detailed 2 or 4 bits.

d.(ii) Some candidates used 6^2 instead of 2^6. 63 was also a common incorrect answer.

QUESTION 14

(a)

Colour	name	Hex Code	RGB
	crimson	DC143C	(220,20,60)
	tomato	FF6347	(255,99,71)
	dark golden rod	B8860B	(184,134,11)
	golden rod	DAA520	(218,165,32)
	dark olive green	556B2F	(85,107,47)
	green yellow	ADFF2F	(173,255,47)
	sea green	2E8B57	(46,139,87)
	blue violet	8A2BE2	(138,43,226)
	dark slate blue	483D8B	(72,61,139)
	slate blue	6A5ACD	(106,90,205)
	dark orchid	9932CC	(153,50,204)
	medium orchid	BA55D3	(186,85,211)
	medium violet red	C71585	(199,21,133)
	deep pink	FF1493	(255,20,147)
	sienna	A0522D	(160,82,45)

Which colour is (85,107,47)? 85 = 01010101, 107 = 01101011, 47 = 00101111

Colour is... 0101 0101 0110 1011 0010 1111 = 556B2F = "dark olive green"

b.(i) Each bit is multiplied by the next power of two down starting at 2-1.

$1 \times 2^{-1} + 0 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} = 0.5 + 0.0625 = 0.5625_{10}$

b.(ii) 0.5 is equal to 0.1000_2 and 0.03125 is equal to 0.00001_2. This means the 4 bit mantissa for 0.5 does not have enough bits to be able to add 0.03125 and so it would not change the value of 0.5.

(c) If the programmer has a table to code the characters W, S, A, D the codes could be 00, 01, 10 and 11 so using 2 bits. A group would be 3 x 2 = 6 bits. The 11 groups would be 11 x 6 = 66 bits.

The maximum number of groups is 11 and this will require 4 bits (16 possible values) to represent. This brings the total to 4 + 66 = 70 bits.

EXAMINER COMMENT

(a) A simple question with almost all candidates getting this correct.

b.(i) Again well answered. Many candidates gave the answer as a fraction, which was accepted.

b.(ii) Good answers showed that $.03125_{10}$ was $.00001_2$, and hence was 'lost' in the small mantissa.

(c) Many combinations were presented, and those that showed some good thought gained partial recognition, despite not using the least storage.

QUESTION 15

(a) If an Animated Gif stores 44 frames for an image and each Image Frame requires 36720 bytes for the pixel colour information what is the file size of the Gif file in bytes? Show your calculation.

File size = 800 + 44 x (18 + 36720) +1 = 1617273 bytes.

(b) The while loop will continue until the value of p reaches the maximum of 127 (01111111_2) and then when one is added it becomes 10000000_2 = -128 (Two's complement negative value). Since p is negative the loop will exit.

On the next line, one is subtracted from 10000000_2 giving 01111111_2 which is final value of 127.

When the value of p exceeded 127 it should have produced an integer overflow error causing the program to terminate.

EXAMINER COMMENT

(a) Generally well answered by those that attempted it. A small number just gave a (wrong) answer without any explanation, which gained no points.

(b) Very few candidates explained the complete process, where p became -128, then adding 1 returned it to 127. Many pointed out the potential error message (although this doesn't occur in practice), and were given partial marks for this.