

2023 ASSESSMENT REPORT

ITC315118 COMPUTER SCIENCE

Key points:

- Many students left entire questions blank. It's important that students remember to attempt all questions as sometimes the easier parts of the exam are contained in the second or third question within a booklet.
- Overall, the most difficult part of the exam was Section 3. To obtain a “C” rating on criterion 3, students are expected to be familiar with the basics of working with objects and their methods, including how to declare and instantiate an object given a class definition.

Section A

Question 1

a) 4

b) 136

c) Insert after line 8

```
if (crepes > 12)
    crepes = 12
endif
```

d) Replace lines 18 and 19 with:

```
else if (crepes < 9)
    set eggs = 2
else
    set eggs = 3
endif
```

Comments

This question was generally well answered by most students. The best responses to 1c and 1d included a clear indication of where the modifications need to go (e.g. “after line 8”) and whether existing lines in the algorithm were to be replaced or amended. They also ensured that any existing parts of the algorithm continued to work correctly.

Question 2

a) Insert after line 8:

```
if (roll < 1)
    set roll = 1
if (roll > 6)
    set roll = 6
```

b) Insert after line 16:

```
second_last_roll = last_roll
last_roll = roll
```

c) Replace line 11-14:

```
if (count <= 10 and last_roll == second_last_roll)
    score = score + roll * 2
else if (count < 10)
    score = score + roll
```

Comments

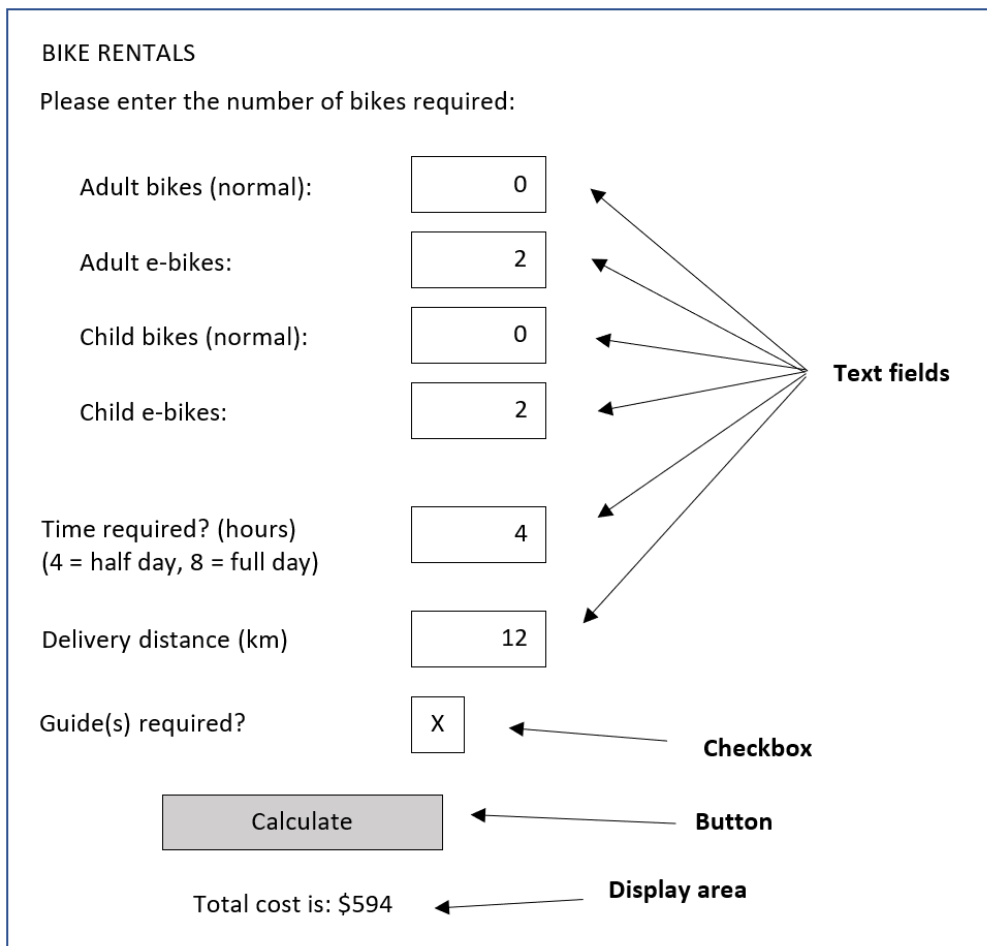
a) Successful responses validated the value of the roll *after* it had been extracted from the text field. Weaker responses confused the **roll** variable with the **“roll” text field**.

b) Generally well answered by most students.

c) This was particularly challenging for many students with only ~10% obtaining full marks. A common approach that gained partial marks was to subtract 1 from **count** if the two previous rolls were equal – potentially leading to multiple bonus rolls. The best responses ensured that only one bonus roll would score double. Some responses went a step further and modified the algorithm so that the double bonus would only apply when the two previous rolls were equal, whether it was the extra roll or not. Few responses seemed to recognise that because the roll was when **count** = 0, therefore the eleventh roll (i.e., the extra roll) would be when **count** = 10.

Question 3

a)



b)

```
Initially
  set adult = 0
  set adultE = 0
  set child = 0
  set childE = 0
  set totalBikes = 0
  set duration = 0
  set distance = 0
  set deliveryCost = 0
  set guidesRequired = 'N'
  set guideCost = 0
  set totalCost = 0
```

When a value is entered in the adultBike text field
set adult to value in the adultBike text field
call the "update cost" method

(Repeat for each of the text fields and check box)

When the "update cost" method is called

```
totalBikes = adult + adultE + child + childE

if (totalBikes > 36)
  display "too many bikes"
else
  // base cost
  if (duration < 4)
    bikeCost = (adult*15 + adultE*25 + child*10 + childE*20) * duration
  else if (period = 4)
    bikeCost = (adult*50 + adultE*100 + child*35 + childE*60)
  else
    bikeCost = (adult*80 + adultE*160 + child*55 + childE*100)
  endif

  // delivery cost
  vehicles = round up (totalBikes / 18)
  deliveryCost = (50 + 2 * distance) * vehicles

  // guide cost
  if (guidesRequired = 'Y')
    guides = round up (totalBikes / 12)
  endif
  guideCost = (duration * 50) * guides

  // total cost
  totalCost = bikeCost + deliveryCost + guideCost
  display totalCost

end if
```

Comments

a) Markers were looking for user interfaces that included all the required inputs and clearly indicated which elements were text fields, buttons, etc. The best responses made it very clear to the user what information was required.

b) The best responses to this question included:

- successful use of the initially/when model, including an algorithm that would work regardless of the order the buttons or text fields are used
- calculation of base cost taking into account the types of bikes and duration of the booking
- calculation of additional costs for delivery and guides
- handling the maximum number of bikes.

Section B

Question 4

a) i. $b = 7, c = 2$

ii. 1.25

iii. 16

Explanation: y will continue to be halved and loop will terminate when y is 16 as 16 is not greater than 30.

b) i. Error 1: need to declare loop counter variable x

```
for(int x = 1; x <= 5; x++)
```

Error 2: need brackets around entire expression

```
if((x % 2 == 0) && (d < 4))
```

ii. 16

Explanation: e (6) is greater than 0 so e becomes 18.

e (18) is greater than 10 so e becomes $18 - 2 = 16$.

iii.

| i | f |
|---|---|
| | 3 |
| 0 | |
| 1 | |
| 2 | 5 |
| 3 | 7 |
| 4 | 9 |
| 5 | |
| 6 | |
| 7 | |

Final value of f : 9

c)

```
public double question4c(double cost, int margin){  
    salePrice = cost + (cost * margin)/100;  
    return salePrice;  
}
```

Comments

- a) Completed correctly by most candidates; very few omitted an answer for each question.
- b) Most candidates successfully completed parts (ii) and (iii). Only a third managed to successfully complete part (i). The issue seemed to be unfamiliarity with this new type of question.
- c) Only half managed to successfully complete this question successfully. Those who didn't, generally did not attempt it at all.

Question 5

- a) i. Value of g: 3

Explanation: $15.0 / 4.0 = 3.75$ (int) $\rightarrow 3.75 = 3$

- ii. Value of h: 47

Explanation: (int) ('3') is 51 \rightarrow (int) Math.round(3.6) is 4 $\rightarrow 51 - 4 = 47$

- b) Final values of array card:

| card | | | |
|------|---|---|---|
| 0 | 1 | 2 | 3 |
| B | L | R | J |

- c) i. Final values of array x:

| | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 1 |
| 2 | 3 | 4 | 1 | 2 |
| 3 | 4 | 1 | 2 | 3 |
| 4 | 1 | 2 | 3 | 4 |

- ii. Array index out of bounds

Comments

- a) i. Completed correctly by most candidates.

ii. Most candidates failed to recognise that (int) ('3') returns the ascii value of character 3 which is 51 rather than 3, most recognised that (int) Math.round(3.6) returned 4 so a common answer was -1 instead of the correct answer 47.

- b) Around half of all candidates completed this with varying degrees of success. Some did not recognise that card.length - 1 meant that the last element 'j' remained unchanged. Around half the candidates did not attempt this question.

- c) Completed correctly by 40% of the candidates. Some candidates were unable to successfully complete the table. In part (ii) many candidates identified that an error would occur but were unable to correctly identify the error, in this case partial marks were awarded.

Question 6

a) The index of the smallest element was 4

The two elements at indices 1 and 6 are 6 and 2 add up to 8

The two elements at indices 2 and 5 are 3 and 5 add up to 8

The two elements at indices 3 and 4 are 7 and 1 add up to 8

1 2 3 4 5 6 7

b) **list** is passed by reference. This means that any changes made to **list** in the method will persist upon completion of the method as the actual parameter is pointing at the same memory location as the formal parameter.

c) Array index out of bounds. To fix this, use **list.length** instead of a hardcoded value.

d) (Changes required in bold)

In the **sort()** method, add parameter **ascending**:

```
public static void sort(int[] list, boolean ascending)
```

then add

```
if (ascending) {  
    if (list[j] > list[j+1]) {  
        int temp = list[j];  
        list[j] = list[j + 1];  
        list[j+1] = temp;  
    }  
} else {  
    if (list[j] < list[j+1]) {  
        int temp = list[j];  
        list[j] = list[j + 1];  
        list[j+1] = temp;  
    }  
}
```

In main method

```
findAndPrintMin(list);  
findAndPrintPairs(list, target);  
sort(list, true);  
showList(list);  
sort(list, false);  
showList(list);
```

Comments

a) 25% of candidates received half marks of better. 19/140 candidates were awarded full marks.

b) Only 18% of candidates were able to gain full marks for this question by explaining that list was passed by reference and any changes made would be retained on completion of the method. A further 14% provided a partial solution and were awarded half marks.

c) About 40% of candidates attempted this question. Most correctly identified the error. Better answers recommended using `list.length` rather than just modifying the value from 7 to 8.

d) Around 28% of candidates attempted to answer this question with varying degrees of success. 8 candidates were awarded full marks for addressing all the question's requirements including all of the method alterations and new method calls.

Section C

Question 7

a) `Cat c = new Cat ("Tom", 6, "male");`

b) `String`

c)

```
public void setBreed(boolean pureBred) {
    isPureBred = pureBred;
}

public boolean getBreed() {
    return isPureBred;
}
```

d)

```
c.setBreed(true);

boolean status = c.getBreed();
```

Comments

a) The constructor's parameters were `String`, `int`, and `String`. Some students did not select the correct choice.

b) Most students answered this question correctly.

c) Most students answered this question well, but many did not fully understand the requirements of this question. Some students had a boolean parameter in the setter and assigned it the value of ***isPureBred*** rather than use it to assign a value to ***isPureBred***. A small number of students tried to use `true` as a boolean variable when it is a literal [has a defined value] and cannot be used as a variable.

d) Some students claimed to not understand this question, some discussed how they guessed the methods could be used. Most answers showed understanding of calling methods.

Question 8

a) `TimeTable student = newTimeTable();`

b) `student.setSubject(3, 2, "Computer Science");`

c)

```
public String getSubject(int day, int period) {  
    return ttable[day][period];  
}
```

d)

```
public void getDaySubjects(int day) {  
    System.out.println("Day" + day)  
    for (int i = 1; i < 4; i++){  
        System.out.println(ttable[day][i]);  
    }  
}
```

e)

```
for (int i = 1; i < 6; i++) {  
    get getDaySubjects(i)  
}
```

Comments

a) Many could not declare and instantiate the class.

b) Many did not find the method and many of these did not know how to call the method.

c) Many did not know how to solve this question. There were some erroneous guesses.

d) The best responses used a for loop to retrieve and print the values. A few got their assignment of student timetable data reversed.

e) Many did not write a valid method in part (d) so could not do part (e).

Question 9

```
public class Score {

    int division;
    int score;
    int highestScore;
    int lowestScore;
    int spread;

    public Score(int score, int division) {
        this.division = division;
        this.score = score;
        highestScore = score;
        lowestScore = score;
    }

    public void setSpread() {
        spread = highestScore - lowestScore;
    }

    public int getSpread() {
        return spread;
    }

    public void setScore (int score){
        if (score > highestScore) {
            highestScore = score;
        }
        if (score < lowestScore) {
            lowestScore = score;
        }
        setSpread()
    }
}
```

Comments

- a) The best responses showed an understanding of class structure, including declaration of class variables, the constructor, and methods. In addition, methods were clearly set out between the class's opening and closing brackets.
- b) The division could be handled by passing it to the class via the constructor. Alternatively, the division could be handled by passing it as a variable to the methods and having corresponding variables in the methods that depend upon the value of Division.
- c) Setting **score**, **highestScore**, **lowestScore** and **spread** was well handled.

Section D

Question 10

- a) i. $\sim(A \wedge B) \vee B$

ii.

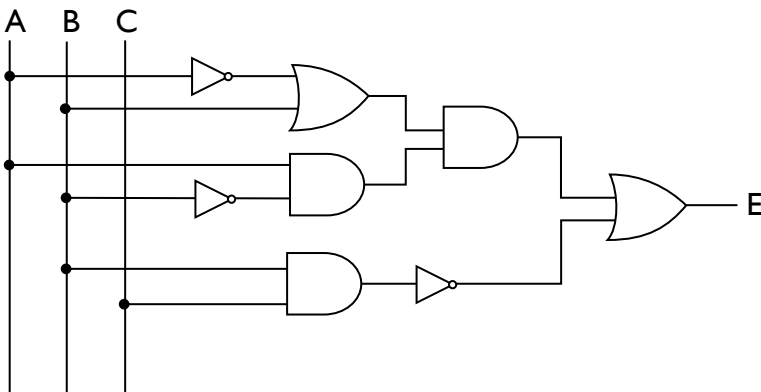
| $\sim A \vee B$ |
|-----------------|
| T |
| T |
| F |
| T |

iii. A

b) i.

| A | B | C | $\sim A$ | $\sim A \wedge C$ | $A \wedge B$ | D |
|---|---|---|----------|-------------------|--------------|---|
| 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 1 | 1 |

ii.



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

c) 0002

Comments

a) These questions were very well answered, with more than 80% of students supplying the correct answer in each question.

b) i. The truth table was very well completed, with only the occasional errors.

ii. Most students drew an accurate circuit diagram. The only common error was drawing $\sim B \wedge \sim C$ instead of $\sim(B \wedge C)$.

iii. This question was very well answered, with the most common error being students mixing up their “P”s and “R”s.

c) Only a third of students received full marks for this question. Some students put their working for this question where it was not required. This was pleasing to see as it meant they could receive part marks for incorrect answers.

Question 11

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

ii)

$$\begin{aligned} E &\equiv (A \vee \sim A) \wedge \sim(\sim B \vee \sim C) \\ &\equiv T \wedge \sim(\sim B \vee \sim C) && \text{(L14)} \\ &\equiv \sim(\sim B \vee \sim C) && \text{(L1, L23)} \\ &\equiv \sim\sim B \wedge \sim\sim C && \text{(L10)} \\ &\equiv B \wedge C && \text{(L6)} \end{aligned}$$

b) i. Some operations need to specify three registers (e.g. ADD adds the contents of two source registers and places the result into a destination register), whereas others need to specify one register and one memory location (e.g. LOAD copies the value contained in a memory location in a register). Because there are 16 registers, only four bits are required to identify each register. Because there are 256 memory locations, 16 bits are required to address each of these.

ii. The Von Neumann architecture introduced the idea of storing program instructions in the same memory as data. This marked the transition from fixed-function machines to stored-program computers. Programs became a sequence of instructions that could be modified more easily than physically rewiring the machine.

iii. In this architecture, the same data bus is used to transfer program instructions and data. This means that program execution may be limited by speed of data transfer between the memory and the CPU.

Comments

a) i. Many students did not utilise the fact that Karnaugh map “circles” are able to overlap each other and they can wrap-around the edges. This led to many inefficiencies in solutions, but these were still awarded full marks. The circle $\sim A \wedge B$ was entirely redundant, but was still included by many students. A number of students would write $\sim(A \wedge D)$ rather than $\sim A \wedge \sim D$ – this is simply not correct!

ii. Almost all students made a good attempt at this question. The most common error was not switching \vee to \wedge when using De Morgan’s Law. A number of students failed to recognise $T \wedge B \wedge C$ could be re-written as $B \wedge C$ using law 23.

b) i. This question was very well answered.

ii. Again, this question was very well answered, with a wide range of answers accepted as correct. The main difference was that now instructions could be fed to the computer using storage rather than rewiring the machine.

iii. The critical point here is that the same bus is now being used to transfer both instructions and data, which cannot be done at the same time, reducing efficiency. Around half of all students were awarded full marks for this question.

Question 12

a)

| Memory Address | Contents | Pseudocode | Explanation |
|----------------|----------|----------------------------------|--|
| 01 | 0001 | data | Used for variable x |
| 02 | 0000 | data | Used for variable y |
| | | | |
| 10 | 8101 | $R[1] \leftarrow \text{mem}[01]$ | Set register 1 to value of x |
| 11 | 7304 | $R[4] \leftarrow 4$ | Set register 4 to value of 4 |
| 12 | 7507 | $R[5] \leftarrow 7$ | Set register 5 to value of 7 |
| 13 | C116 | if (R[1] == 0) pc ← 16 | if (x=0) goto line 16 |
| 14 | D116 | if (R[1] > 0) pc ← 16 | if (x>0) go to line 16 |
| 15 | 0000 | HALT | |
| 16 | 2A41 | $R[A] \leftarrow R[4] - R[1]$ | $R[A] = 4 - x$ |
| 17 | DA19 | if (R[A] > 0) pc ← 19 | if R[A] > 0 then x must be less than 4 |
| 18 | 0000 | HALT | |
| 19 | 9502 | $\text{mem}[02] \leftarrow R[5]$ | Set y = 7 |
| 1A | 0000 | HALT | |

b)

| PC | Contents | R[1] | R[3] | R[4] | R[6] | mem[06] |
|----|----------|------|------|------|------|---------|
| 10 | 8101 | 1 | | | | |
| 11 | 8303 | | 3 | | | |
| 12 | 8404 | | | 7 | | |
| 13 | 8606 | | | | 0 | |
| 14 | C318 | | | | | |
| 15 | 1664 | | | | 7 | |
| 16 | 2331 | | 2 | | | |
| 17 | C014 | | | | | |
| 14 | C318 | | | | | |
| 15 | 1664 | | | | 14 | |
| 16 | 2331 | | 1 | | | |
| 17 | C014 | | | | | |
| 14 | C318 | | | | | |
| 15 | 1664 | | | | 21 | |
| 16 | 2331 | | 0 | | | |
| 17 | C014 | | | | | |
| 14 | C318 | | | | | |
| 18 | 9606 | | | | | 21 |
| 19 | 0000 | | | | | |

Comments

- a) Almost half of candidates did not attempt this question, which was disappointing to see. Of those who did, most of them gained near-full marks. The most common errors were not handling correctly the cases where $x = 0$ or $x = 4$, or implementing $x \geq 0$ *or* $x < 4$ rather than “and”.
- b) This question was in many ways easier than part (a) but was left by almost the same number of students. Again, this was disappointing to see. Common errors were believing that the program was calculating “4+4+4” rather than “7+7+7” and not including every single instruction in the trace. More students gained full marks in this question than (a). Both answers in hexadecimal or decimal were accepted here.

Section E
Question 13

- a) i. 1000 10102
 ii. 53
 iii. 100

b)

$$\begin{array}{rcccccc}
 & & 1 & \boxed{1} & 1 & \boxed{0} & 1 \\
 + & 1 & 0 & 0 & 1 & 1 & \\
 \hline
 1 & \boxed{1} & 0 & \boxed{0} & 0 & 0 &
 \end{array}$$

- c) i. 0110 0011₂
 ii. 1010 11002
 iii.

$$\begin{array}{r}
 94 \quad 0110\ 0011 \\
 +(-84) \quad +\ 1010\ 1100 \\
 \hline
 =\ 0000\ 1111 \quad \rightarrow\ 15_{10}
 \end{array}$$

- d) 0100 1010 0111₂
 e) $0.11_2 = 2^{-1} + 2^{-2} = 0.75_{10}$

Comments

- a) Correctly answered by nearly all students.
- b) Well answered by most students. Part marks were awarded for each of the boxes that were correctly filled.

c) Parts i and ii were well answered by most students. Part marks were given to students who found the one's complement in part ii.

d) Well answered by most students.

e) Generally well answered. Students who stated that the two numbers were simply binary and decimal equivalents received no marks.

Question 14

a) $0.1101_2 = 2^{-1} + 2^{-2} + 2^{-4} = 0.8125_{10}$

b) 7 colours can be represented by 3 bits ($2^3 = 8$ options)

10 numbers can be represented by 4 bits ($2^4 = 16$ options)

Therefore $3 + 4 = 7$ bits are needed to represent each card.

c) i. $1024 = 1 \times 2^{10} = 0.5 \times 2^{11}$

Sign bit = 0 (positive)

Exponent = $11_{10} = 01011_2$ (in 5-bit two's complement)

Mantissa = $1\ 000\ 000\ 000_2$

16-bit floating point representation = $0\ 01011\ 1000000000$

ii. $0.125 = 1 \times 2^{-3} = 0.5 \times 2^{-2}$

Sign bit = 0 (positive)

Exponent = $-2_{10} = 11110_2$ (in 5-bit two's complement)

Mantissa = $1\ 000\ 000\ 000_2$

16-bit floating point representation = $0\ 1110\ 1000000000$

A double is stored as an approximation. Testing for equality on floating point numbers can often lead to issues as rounding (approximation) errors can occur. By allowing for a small margin of error, the code will work as intended.

Comments

a) Well answered by most students.

b) This question received a wide range of answers with approximately 1/3 correct, 1/3 correct and 1/3 partially correct. Part marks were awarded if the minimum number of bits was calculated for the colours or the numbers. No marks were deducted if students went on to calculate the storage required for the whole deck of cards.

c) This question troubled most students. Part marks were awarded if students obtained one or two of the sign bit, exponent and mantissa. Part marks were also awarded if the students obtained a correct answer but had not normalised the mantissa.

d) This question was poorly done. Part marks were awarded where students explained the purpose of the `Maths.abs()` function or where they mentioned that an approximation was involved without explaining the consequence of this approximation.

Question 15

a) First name, middle name, surname, and nationality → $50 \text{ chars} \times 16 \text{ bits/char} = 800 \text{ bits}$ each, or 3200 bits in total.

Date of birth, Date of Issue and Expiry Date →

- 5 bits to represent 31 days,
- 4 bits to represent 12 months
- 8 bits to represent 256 years (1901 to 2156)
- 17 bits each, or 51 bits in total

Gender → 2 bits to represent 3 options

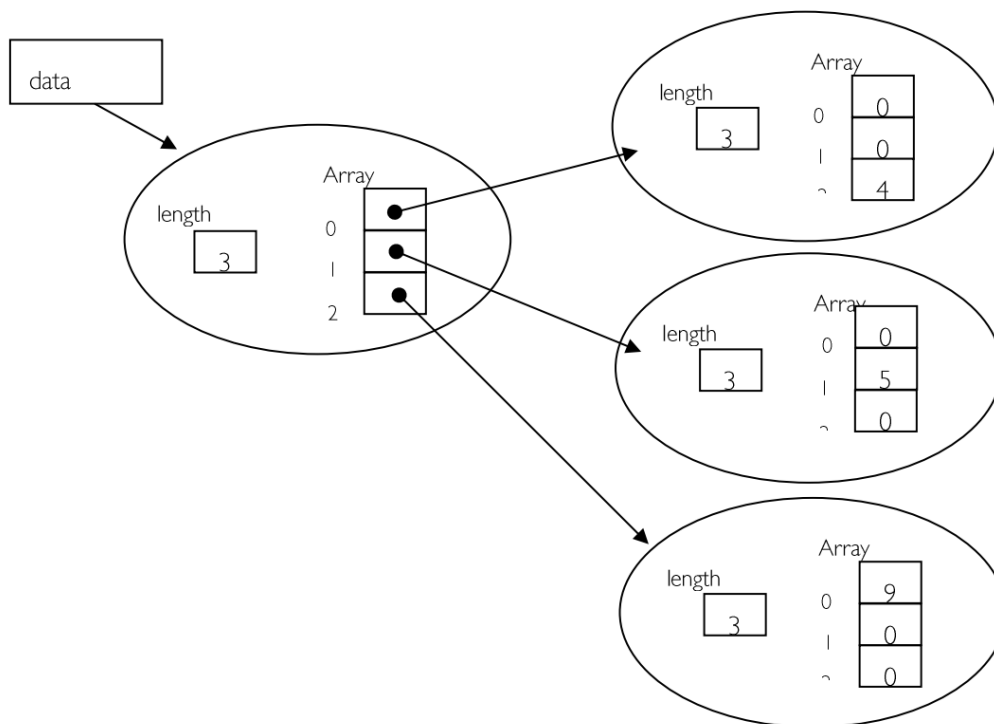
Document Number → 5 bits to represent capital letter and 20 bits to represent numbers from 000000 to 999999

Image → $150 \times 150 \times 24 = 540\,000$ bits

Total = $540\,000 + 3\,200 + 51 + 3 + 25 = 543\,279$ bits

b) A byte can vary from -128 to 127 so the condition $i < 128$ is always true. When you add 1 to 127 it overflows and becomes -128 and so on in an (infinite) loop.

c)



Comments

a) Students provided a wide range of answers. Marks were generally given where students stated the assumptions they had made, provided that the approach of the students didn't result in a significant increase in the storage requirements. Students who stored all symbols (numbers and letters) as 16-bit Unicode were deducted marks as were students who used ASCII, given that the question explicitly asked for characters to be stored as 16-bit Unicode. A different number of bits to code years was acceptable as long as students explained their choice.

b) Most students struggled with this question with very few realising that the code would result in an infinite loop. Part marks were awarded if students suggested that the code would loop from 0 to 127 and then stop.

c) Generally well answered. Marks were awarded for correctly adding lengths to each component of the array, for correctly changing the three elements and for showing zeroes for the remaining elements.