COMPUTER SCIENCE

Paper 0478/12
Paper 1 Theory

Key messages

Candidates often gave appropriate responses to the questions and justified their responses when required. Some candidates did not answer the question they were given, for example, giving uses instead of features, or describing translators instead of explaining when it is best to use each.

General comments

Candidates need to carefully read the questions to make sure they are giving answers that are appropriate in the context. Their answers should also take these contexts into account, making sure that the points they make are relevant to the context.

Comments on specific questions

Question 1

(a) Many candidates were able to correctly convert the binary number to denary. Some candidates wrote their answer in the working space, when they have finished candidates need to make it clear what the intended answer is.

(b) Many candidates answered this question correctly. A common error was converting to denary instead of hexadecimal. Some candidates had the C and 0 in the reverse order, i.e. 0C

(c) Many candidates were able to answer this question accurately. Some candidates converted A to 11 instead of 10.

(d) This question was often answered accurately. Some candidates wrote a binary number in two bytes instead of answering the given question.

Question 2

(a) Many candidates answered this question well with the most common answer being JPEG, but TIFF and PNG were also often given.

(b) Some candidates gave a description of how a camera takes a photograph and stores this in a computer, and some candidates gave a description of secondary storage. The photograph response often met some criteria when the candidates described the use of pixels.

Some responses require further precision, for example, each colour has a binary value could mean that two colours have the same binary value. Likewise, a pixel has a binary value that needs to be extended to what this binary value represents this is because the candidate is otherwise identifying the image is stored in binary without what the binary values mean.

(c) (i) Some responses gave a clear indication that the file size is reduced. Some candidates used the same terminology as in the question without expansion, for example, the data is reduced without stating what this means. A common error was giving an example of compression, for example, lossy compression, instead of what data compression is.

A common misunderstanding is that a compressed file is 'easier' to send. Easier indicates that there are fewer steps for the user to follow. The process is the same whether the file is compressed, or not compressed. It will arrive at its destination faster but for a user, it is no easier. It might be harder if the file is, for example, zipped and the receiver has to then unzip it before it can be used.

(ii) This question was often answered well. Common answers included email having a limit in file sizes and that the file would take less time to arrive at the destination. Some answers gave a definition of data compression, i.e. reducing the size of the file, without linking this to the scenario of why it is needed when emailing a file.

(iii) The stronger responses gave a clear description of the steps involved in parity including how errors are detected. Most candidates described a parity check whilst some candidates described a parity block check.

A common misunderstanding is that the parity bit indicates if the byte should be odd or even. For example, if the parity bit is a 1 then the parity is even. This is incorrect, the parity bit is set according to the number of 1 bits and is a 1 or a 0 depending on which is needed to make that number odd or even.

(iv) Many candidates were able to identify a suitable method. A common error was giving check digit as a method used during the transmission of data; it is used on the input of data to a computer, not in the transmission between computers.

Some candidates gave descriptions of where the users were performing actions, for example, the user was calculating the checksum, or the user was requesting the data to be sent again when there was an error. The error detection methods are automated and it will be the error detection method, or the device making the request, not the user.

Question 3

(a) This was often answered well with candidates correctly identifying the logic gates.

(b) Many candidates were able to draw an accurate logic circuit for the statement. A common error was mixing up the B OR NOT C and drawing NOT (B OR C) instead. When candidates are drawing crossing lines it is helpful to make it clear where each line goes, for example showing it jumping over another line. In some responses, it was not clear which inputs were leading into each gate.

Question 4

(a) (i) This question was answered well by many candidates who were able to correctly complete the description.

(ii) This question was often answered well with many candidates giving suitable input devices. Some candidates gave a repeated device from **part (a)**, i.e. a mouse or keyboard, and some candidates gave output devices such as printers.

(b) There were mixed responses to this question. Some candidates repeated the requirements, for example, that they could watch high-definition films, without expanding on what HD meant.

The most common answers included a higher resolution and greater contrast, or more vivid colours.

A common error was that it could have more colours, whilst the colours will be more vivid there is not a wider range available.

Another common error was stating it was lighter weight and easier to attach to a wall, which is true for a DLP and not an LCD.

(c) Many candidates were able to identify a suitable storage device. Some candidates identified USB which is not a storage device on its own, additional information such as USB memory stick is required.

When justifying their devices candidates would often have benefitted from being more precise, for example instead of stating they are cheaper, identify that they are cheaper per GB. This is because each type of storage device can come in many formats and some may cost less than others.

Question 5

(a) Some candidates were able to give suitable examples of both structure and presentation, commonly tables and headings for structure and colour and font styles for presentation.

Some responses required further content, for example, text for presentation is not clear if it is the font type of the text, the actual words, the location of the text, etc.

(b) Many candidates gave descriptions of what each type of address is used for, why they are used, etc. instead of the features. The most common correct answers included the indication of hexadecimal being used. Fewer were able to give additional features.

(c) The threats needed to be relevant for a web server, this limited the range of threats that could be given. For example, phishing cannot be carried out on a web server.

Where a suitable threat was given candidates often gave suitable impacts, most commonly the loss of data. A common inaccurate software to limit a threat was anti-hacking, which is not precise enough for a firewall or proxy.

Question 6

There was a mix of responses to this question. Many candidates were able to identify how the data is transferred from the sensor to the microprocessor. Fewer candidates were able to explain how the timer would work, with many candidates suggesting that every two minutes the data would be analysed, instead of the data being constantly analysed and a timer resets each time movement was detected.

Question 7

(a) (i) Some candidates gave a description of an interpreter and what it does, instead of answering the question which was when it is most appropriate for it to be used. Some candidates gave suitable reasons for its use but did not indicate when it was being used, for example, that it is easier to debug, without indicating if it was used during the development.

(ii) As with **part (a)(i)** some candidates described how a compiler works instead of when it should be used.

(b) This question was answered well with many candidates giving suitable features of both freeware and free software. A common error was mixing up freeware and free software, primarily identifying that freeware can be modified but free software cannot be modified.

COMPUTER SCIENCE

Paper 0478/22

Paper 2 Problem-solving and Programming

Key messages

Candidates who had completed the tasks for the pre-release (golf scoring) were able to provide answers for ***Section A*** that showed a good understanding of the tasks undertaken. Candidates, who read each question carefully and answered the question, as set on the paper, performed better than those who had memorised their solution and used all of that information without considering what information needed to be included in their answer. For example, **Question 1(d)** asked for an explanation about how the program completed part of **Task 3.** This question only required the candidate to explain how the program completed that part of the task rather than explain how the program completed the whole task.

Candidates should take care when declaring variables, constants, and arrays to ensure that the identifier declared could be used in a program. Identifier names must not contain spaces. Once declared the same identifier name should be used throughout the answer.

Trace tables should be clearly completed in ink, not in pencil with alterations made in ink because both answers are visible when the answer is scanned.

General comments

Most candidates attempted all questions in both sections.

Comments on specific questions

Section A

Question 1

(a) Most candidates correctly stated a suitable variable to use in **Task 1**. Many candidates explained why a variable was used. Stating a suitable constant for use in **Task 1** proved more of a challenge as several candidates chose identifiers that could not be used as a constant because the task specified that the value was to be input by the user, for example `Par` or `NumberOfHoles`.

A common error seen for both the variable and the constant was identifying what it was used for rather than why a variable or constant was chosen.

(b) Most candidates gave a description of how their program set the values in the array(s) used to store the players' scores to zero at the start of the round. A common error was to include programming code without an explanation.

(c) Generally, well answered. Algorithms were usually written in pseudocode or program code; a few flowcharts were seen. Most candidates showed most of the steps required for **Task 2**. Common errors in the programs included, entering all scores for one player at a time rather than entering the score for each player at every hole played, storing these scores in a variable instead of an array, and lack of prompts and/or verification for entry of scores.

(d) A few candidates did not attempt this part of the question. Well answered by those candidates who explained how their program completed the part of **Task 3** that identified the winner and displayed the winning score relative to par. Common errors included writing about the whole of **Task 3**, including programming code without an explanation, or identifying the player with the highest number of strokes instead of the player with the least number of strokes as the winner stated in the scenario.

Section B

Question 2

(a) A few candidates did not attempt this part of the question. Generally, well answered when attempted. Common errors seen, included using `OR` instead of `AND`, `<=` instead of `<`, or omitting the variable `Age` for `AND Age < 12`.

(b) Generally, well answered. Most candidates wrote appropriate, well-constructed pseudocode statements using the given variable `CountUnder7`. A common error seen was to only include the pseudocode statements for counting and omit those required for outputting the number of students under the age of 7. Other errors seen were the use of `<=` instead of `<` or the use of an incorrect variable name.

Question 3

Most candidates could match each validation check with the most appropriate description.

Question 4

(a) The full range of marks was awarded with many candidates showing the skill of completing a trace table. Common errors seen included not realising that the last two values of the input data would not be used or including extra values in the output column.

Some trace tables were completed in pencil with alterations made in ink, this made identifying correct answers difficult because both answers were clearly visible when the answer was scanned.

(b) Many candidates correctly identified that the number of factors was found, better responses described the purpose more fully as counting and displaying the number of factors of the number input. A common error was to describe the purpose as finding odd, even, or prime numbers.

(c) (i) A few candidates did not attempt this part of the question. Most candidates described at least one aspect of the problem better responses gave a fuller description. For example, 'The value of `D` will become zero before it is tested, so the test is `D=1` will always fail resulting in an endless loop.'

(ii) A few candidates did not attempt this part of the question. Explaining the changes required to the flowchart proved challenging for many candidates. Candidates needed to identify where in the flowchart the change needed to be made and the change required.

Question 5

Many candidates gained good marks for this question showing an understanding of the workings of a query-by-example grid.