

Marking Feedback Comments are formatted as this paragraph — light yellow background with black border. There may also be a green tick ✓ where you have made a good point.

Any grammar, spelling and punctuation comments are highlighted, and the correction given. I hope that there is no confusion arising from formatting changes to your marked document caused by my feedback comments, but you have of course got your original document for reference. The marks for individual questions are available in the Assessment Summary (PT3) file which also contains other information that will help you improve in future assignments.

Please read my comments carefully and let me know if you have any questions about this or any other aspect of your assignments.

Well done for including a header showing your name, PI number and assignment details. Please insert a footer showing the page details as well.

Please start each question on a new page.

‘CTRL and Enter’ inserts a page break; this is better than inserting blank lines as the question will always be at the top of a new page, even if you add or delete lines from the previous pages later.

Question 1**(a)**

(i) Because **unsigned numbers** encodes a non-negative number in the range of **(0 to 4294967295)** and **signed numbers** encodes integers from **(-2147483648 to 2147483648)**

To answer the question, you need to explain that the count can't be negative, and a 32-bit unsigned representation can hold a larger positive number than it is possible to store in a signed 32-bit number.

(ii) This expression of the largest number could be represented as $2^{32-1} = 4294967295$

The largest number than can be stored in a 32-bit binary representation is $2^{32} - 1$. Be careful that -1 is not a superscript.

(iii) In this case It might create an **integer overflow** problem, because it is creating a value that is outside of the range that can be represented by the number above.

Correct. The problem of overflow will occur if more than 4,294,967,295 people try to join the group as this will exceed the largest number that can be represented in a 32-bit unsigned binary representation.

(b)

(i) Convert 0.10111000_2

2^0	.	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}
0	.	1	0	1	1	1	0	0	0
0	.	0.5	0	0.125	0.0625	0.03125	0	0	0

$$0.10111000_2 = 0 + 0.5 + 0 + 0.125 + 0.0625 + 0.03125 + 0 + 0 + 0 = 0.71875_{10}$$

A correct answer and clear working.

(ii) Convert 0.10111001_2

2^0	$\square\square.$	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}
0	.	1	0	1	1	1	0	0	1
0	.	0.5	0	0.125	0.0625	0.03125	0	0	0.00390625

$$0.10111001_2 = 0 + 0.5 + 0 + 0.125 + 0.0625 + 0.03125 + 0 + 0 + 0.00390625 = 0.72265625_{10}$$

A correct answer and clear working.

(iii) It would take 20 digits to represent a close approximation of 0.72

The sum would be of the numbers $\frac{1}{2} + \frac{1}{8} + \frac{1}{16} + \frac{1}{32} + \frac{1}{1024} + \frac{1}{4096} =$ which in binary is

$$10111000010100011110 \approx 0.719970703125$$

The closest value can be found by subtracting the decimal values.

$$0.72 - 0.71875 = 0.00125$$

$$0.72265625 - 0.72 = 0.00265625$$

$$0.00125 < 0.00265625.$$

The smallest difference is 0.00125. So, 0.71875 which is represented by the binary number 0.10111000_2 is the closest approximation.

(iv) I couldn't finish this question it was difficult for me to finish it

Changing \$100,000 at an exchange rate of 0.71875 would give

$$100,000 \times 0.71875 = \text{£}71,875$$

Changing \$100,000 at an exchange rate of 0.72265625 would give

$$100,000 \times 0.72265625 = \text{£}72,265.625$$

$$\text{The difference is } 72,265.625 - 71,875 = 390.625$$

So, the new representation means an extra £390.63 (to the nearest penny).

Notice how the unrounded amount is shown. That way, if you make a mistake in rounding, you will still gain some marks.

Ref: Section 1.2.2, Particularly Activity 1.19, p37

(c)

(l)

1 person present			
0 person not present			
Area Manager	Branch Manager	Clerk	Safe can be opened (1) Safe can't be opened (0)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

A correct table. Well done.

(ii)

A	B	A? B	Logic expression
---	---	------	------------------

0	0	1	
0	1	0	NOT A AND B
1	0	0	A AND NOT B
1	1	1	

The question instructs you to use the method in Block 1 Part 1 (Ref: Section 1.3.2). This requires that first an expression is constructed for each row that takes value 1 for $A \neq B$.

A	B	$A \neq B$	Logic expression
0	0	1	NOT A AND NOT B
0	1	0	
1	0	0	
1	1	1	A AND B

These logic expressions are then combined with OR to obtain final expression:
 (NOT A AND NOT B) OR (A AND B)

Ref: Section 1.3.2, Activity 1.28, p53

Question 2

(a) My algorithm

>Draw A frieze Pattern across the page

Set number_of_shapes to 5 (unnecessary)

for sections from 1 to 5 unless you number_of_shapes here

For consistency, you could call the *sections*, 'shapes'

- >> Draw frieze of 8 spokes each
- >> Move forward to draw the next frieze

A good top-level decomposition. Well done for using the notation and indentation as taught in TM112.

(b) A more refined algorithm

> Draw a frieze Pattern across the page

Set the number of shapes to 5

Go to the middle of the screen

For the number of shapes

>> Draw frieze of 8 spokes each

For the 8 sides of the spokes

draw a line of length 40

Make sure that you are specific

go back to centre

turn Right angle 45 degrees

>> Move forward to draw the next frieze

Put the pen up

Move 100 units to the right

Put the pen down

On this program I used a nested loop to make my program a bit cleaner, following the examples of the block 1

Further decompose the top-level decomposition to obtain an algorithm which can be translated directly into code. (So, the lines from the top-level decomposition should still be here.)

Make sure that your algorithm includes all the steps. It should be translated directly into Python code.

(Ref: Section 2.2.4, Page 101)

Remember to indent the processes within each loop as in the following example:

By stepwise refinement of the top-level decomposition, we get:

```
> Draw frieze
for element from 1 to 5
  >> Draw element
  for spoke from 1 to 8
    draw a line of length 40
    go back to centre
    turn 45 degrees left
  >> Move to next element
  take pen off paper
  move forward by 100
  put pen on paper
```

(Ref: Section 2.2.4, Page 101)

Indentation makes it clear which processes occur within the loop.

(c) My code for the frieze pattern

```
# Draw frieze
from turtle import *

# set the thickness of the pen
pensize(1.3) not specified

# start to draw at this position
penup()
goto(-300,0) not specified and does not correspond to your algorithm.
pendown()

for shape in range(5):
    # Draw a spokes

    for sides in range(8):
        forward(40)
        backward(40) backward not taught
backward does not correspond to your algorithm
        right(45)

    # move forward to start position of the next frieze

    penup()
    forward(100)
    pendown()

done() not required and not taught on TM112
```

To demonstrate your understanding of the course materials, you should use the techniques taught in the module. Otherwise, unless you have a reasonable justification, you will lose marks. This also helps to prevent plagiarism.

It is good to see the comments in your code; this makes it easy to follow. Remember to include a comment to explain what the program does. Your code produces 5 shapes of the correct size. However, the starting position should be at the default (0.0). Make sure that you do exactly what is specified, and nothing more.

Example answer:

```
# Draw frieze
from turtle import *
for basic_element in range(1, 6):
    # Draw basic element
    for spoke in range(1,9):
        forward(40)
        left(180)
        forward(40)
        left(180)
        left(45)
    # Move to next element
    penup()
    forward(100)
    pendown()
```

Note: Instead of 'in range (5):', 'in range (1, 6):' can be used.

The Python code 'in range (1, 6):' repeats 5 times using a variable 1 to 5 inclusive.

Whereas 'in range (5):' repeats 5 times using a variable 0 to 4 inclusive.

Sometimes, we want to start counting at one so that we can use the variable in a calculation.

(d) The modified code

```
from turtle import *
```

```
# set the thickness of the pen  
pensize(1.3)
```

```
# the number of shapes was changed from 5 to 7
```

```
number_of_shapes = 7  number_of_elements = 5
```

Make sure that you follow the instructions. Five elements are still required.

```
# start to draw at this position
```

```
penup()  
goto(-300,0)  
pendown()
```

```
for shape in range(number_of_shapes):
```

```
    # Draw a spokes
```

```
    for sides in range(8):  
        forward(40)  
        backward(40)  
        right(45)
```

```
# move forward to start position of the next frieze
```

```
penup()  
forward(100)  
pendown()
```

```
done()
```

You have correctly amended your code to use a variable for the number of elements. However, do take care to read the question carefully as there were 5 elements required throughout the whole question. To demonstrate your understanding of the course materials, you must make sure that you use the techniques taught in the module.

Note: Instead of 'in range (5):', 'in range (1,6):' could be used.

The Python code 'in range (1,6):' repeats 5 times using a variable 1 to 5 inclusive.

Whereas 'in range (5):' repeats 5 times using a variable 0 to 4 inclusive.

Sometimes, we want to start at one so that we can use the variable in a calculation.

Question 3**(a)**

(i) Multi-core processor is a single chip which contains two or more independent processors called cores. Each core performs functions of loading data and instructions into the registers and performing arithmetic or floating-point manipulations.

Instructions can be shared between each of the cores and run at the same time, increasing the program's speed and generating less heating than a **single-core processor**.

(60 words)

A good answer covering the important points within the word limit.

(ii)**Six-Cores CPUs finally overtake Quad-Core in gaming**

The use of quad-core CPUs in the game industry is declining as game companies are now creating their games specifically for six-core CPUs.

(30 words)**Reference**

Islam, Z. (2022) *Six-core CPUs finally overtake quad-core processors on Steam*, *Digital Trends*. Digital Trends. Available at: <https://www.digitaltrends.com/computing/six-core-cpus-finally-overtake-quad-core-processors-on-steam/>

(Accessed: 1 May 2022).

You have carried out some research and provided a reference to your source using the correct format. Well done. Remember to include an in-text citation.

(b)

(i) since 1 byte = 8 bits, 3 bytes is equal to $3 \times 8 = 24$ bits, so each sample contains **24 bits**

correct

(ii) **96kHz/24bit**: $96000 \times 24 = 2304000 \approx 2.304 \times 10^6$ bits per second $2,304,000 = 2.30 \times 10^6$ bits per second (bps) (to 3 s.f.)

A correct calculation. However, the answer is required in scientific notation to 3 significant figures. Remember to explain your working and indicate any rounding in your final answer.

96 kHz = 96×10^3 Hz or 96×10^3 samples/second

Each sample involves processing 24 bits.

The number of bits per second being processed is

$$24 \times 96 \times 10^3 = 2,304,000 \\ = 2.30 \times 10^6 \text{ bits per second (bps) (to 3 s.f.)}$$

So, the number of bits processed while the music is being recorded is 2.30×10^6 bits per second (bps) (to 3 s.f.).

Ref: Section 3.3.3 and Activity 3.14 p.165

Questions (iii), (iv) and (v) I couldn't finish it, I didn't know it.

3biii

Total bits for music = length of music in seconds x bits per second

5 minutes and 10 seconds is $5 \times 60 + 10 = 310$ seconds

$$2.304 \times 10^6 \times 310 = 714,240,000 \\ = 7.14 \times 10^8 \text{ bits (to 3 s.f.)}$$

So, the number of bits processed in 5 minutes 10 seconds is 7.14×10^8 bits (to 3 s.f.)

3biv

Sampling rate is reduced to 24 kHz.

Each sample involves processing 24 bits.

So, the number of bits being processed in 310 seconds is given by:

$$\begin{aligned} 24 \times 24 \times 10^3 \times 310 &= 178,560,000 \\ &= 1.79 \times 10^8 \text{ bits (to 3 s.f.)} \end{aligned}$$

So, in the same piece of music, if the sampling rate is reduced to 24 kHz, the number of bits processed is 1.79×10^8 bits (to 3 s.f.).

Ref: Section 3.3.3 and Activity 3.14 p.165

3bv

The percentage decrease in the number of bits processed is :

$$((714,240,000 - 178,560,000) / (714,240,000)) \times 100 = 75\%$$

[Alternatively, you could use the sample rates:

$$(96 - 24)/96 \times 100 = 75]$$

So, the percentage decrease is 75.00% (to two decimal places).

Ref: Activity 3.2(c), p143, shows a percentage increase calculation, although in a different context.

In your answers to 3b, you showed your working clearly. It could be improved further by explaining what you are calculating. Remember to show every step, including the units throughout, and present your final answer as a full sentence in the specified format. Please see the using numbers booklet for an explanation of significant figures. The slides from my tutorials (posted on the cluster forum) will also help. Please see the post on the tutor group forum which explains how to lay out answers to numeracy questions.

(c)

(I) **The interpreter** works between the program and the computer, and it interprets the programming language code into binary code. The interpreter usually starts right away, however, it runs slowly and lets the programmer see the results as it goes.

Interpreted is when the interpreter translates each instruction in the source code only when it is required for it to be executed.

Advantages

- Start right away
- Let the programmer see the results as it goes and it's easier to correct any errors before the program is executed

Advantage: doesn't require to be compiled in one go (as with compiled programs).

Disadvantages

- It is slow to run than the compiler.

Compilation is the process of translating source code into machine code for execution by the computer.

The compiler piles everything (program) together, it takes more time to start, and it requires extra preparation, but it runs quicker than the **interpreter**.

Advantages

- Run and execute the program quicker

Disadvantages

- It takes more time to start the program
- Requires extra preparation, for example, some extra commands
- It is hard to detect errors before executing the program

Disadvantage: any changes to the program requires the source code to be completely recompiled.

You could improve your answer with separate paragraphs for compilation and interpretation, including the relevant explanation of each at the beginning.

To gain full marks for this question, you must provide an advantage and disadvantage for both compilation and interpretation. So, it is best to focus on answering the question and just give one advantage and one disadvantage of each, using full sentences rather than bullet points.

(Ref B1 P3: 3.5.2)

(ii) **Python** is an example of an interpreted programming language, and **C** is an example of a compiled programming language.

Correct examples

(150 words)

Question 4**(a)****(i)** The input and output are integers value, which is numeric data types

You need to be precise.

There are three inputs, each is an integer.

There is one output: a string.

Ref: p.205, Section 4.2.2, Part 4, Block 1

(ii) only numeric data types, specifically, integers are the values admissible

You need to be precise.

The admissible inputs are day, which is an integer between 1 and 31 (which must be valid, depending on the month), month which is an integer between 1 and 12, and year which is an integer between 1922 and 2010.

(b)

(i) The second conditional statement is missing the **colon** symbol “:” at the end of the statement. ✓

And at line 40 “**age = 2022 – year**”, there is an invalid character “–” and there are indentation errors in the conditional statements.

The syntax errors are:

- a missing colon at the end of the second if statement.
- a missing single quote mark in the print statement.

Note: There were instructions to use the text file provided rather than copying and pasting from the TMA webpage, as the web page contains special invisible formatting characters. I also alerted you to this in my email and on the tutor group forum.

(ii) **Test Number 1, 3 and 4** passed, however, **Test Number 2 and 5** would still have failed because the calculation is still subtracting 2022 from the year input.

Tests 3, 4 and 5 will fail to give the correct age. The question also asks you to explain why.

Each of the three test cases' failure was associated with the line:
if month == 5 and day <= 5. The problem is that the second if condition should be:
month == 5 and day > 5 because age only has to be adjusted if the date of the birthday is after 5 of May.

(iii) those values are the numbers are integers type related to the birthday of the person.

Make sure that you are specific. Tests 1 and 2 test for months before and months after May (the day does not matter if the birthday month is different to the target month). Tests 3, 4 and 5 test on or around the boundary of the day, when the month is the same as the target month.
(Ref: Section 4.2.2, p205)

(c)

(I) because **pattern 4.2** gets the input at the beginning and divides the problem into sub-problems, the program above involves many arithmetic calculations (**formulas**), it gets input data and compares the input data with the data stored in the variables inside the program, those calculations can be transformed into sub-problems, in order to make the problem easier to be implemented.

(60 words)

Pattern 4.2 is more efficient: Once the first if has succeeded there is no need to check further.

(ii) My final algorithm on the pattern 4.2

>> = means process

> = means sub-process

> is the top-level process and >> is a sub-process

>Find age of a person on 5 May 2022

>>initialise ~~variable of~~ Initialisation

>input day of birthday as an integer from 1 to 31

store input in the day variable

>input month of birthday as an integer from 1 to 12

store input in the month variable

>input year of birthday as an integer from 1922 to 2010

store input in the year variable

>> initialize the age variable-age

Age variable is current_year 2022 – year (current year not set)

>> compare the variables calculate age

>if the month is greater than 5

return year – Age is birthday_year – 1

>otherwise if the month is equal to 5 and day is greater than 5

return year - Age is birthday_year - 1

>> Display the final output

print age

Note: User input has not yet been taught and so should not be used in your solution.

Do not use code in your algorithm – it must be written using English.

Your algorithm does not use pattern 4.2.

Please note that indentation is important to indicate the processes that occur when the condition is true.

Example algorithm using pattern 4.2:

>Find age of a person on 5 May 2022

>>Initialisation

Initialise *birthday* to 5

Initialise *birth month* to 5

Initialise *birth year* to 1998

Initialise *current year* to 2022

>>calculate age

if birth month > 5 or (birth month == 5 and birthday > 5)

 age is current year – birthyear – 1

otherwise:

 age is current year – birthyear

>>Print age

Print the *age*

See the Programming and Python quick reference guide in the Resources section:

<https://learn2.open.ac.uk/mod/oucontent/view.php?id=1912071§ion=3.2>

Note: The question states that your algorithm must initialise the day, month, and year to be a valid birthday, which may be your own or any other valid birthday. So, you must state the values.

(d) My python code

Problem: Check how old you are on 5 May 2022

initialise variable

```
day = int(input("day of birthday as an integer from 1 to 31: "))
month = int(input("month of birthday as an integer from 1 to 12: "))
year = int(input("year of birthday as an integer from 1922 to 2010: "))
```

current_year = 2022 redundant code: current_year not used

User input has not been taught yet on TM112.

get input from the user You already obtained the user input above

age = 2022 - year

the comparison check to see if the birthday has yet to occur after the cutoff date

```
if month > 5:
    age = age - 1
```

```
if month == 5 and day > 5:
```

age = age - 1 correct, but does not correspond to your algorithm

the output

```
print("On 5 May 2022 I am " , age , "years old")
```

You have corrected the code and you included comments and sensible variable names. However, remember to include a comment at the beginning of your program to explain what it does and make sure that your code is a direct translation of your algorithm.

Note: Your program must use only constructs that have been taught in TM112 so far, so that you demonstrate understanding of the concepts and mastery of the skills taught in the module materials. If you provide an alternative solution that goes beyond the materials, you may fail to gain these marks.

Example using pattern 4.2:

```
# Problem: Check how old you are on 5 May 2022
# Input: day of birthday, an integer from 1 to 31
# Input: month of birthday, an integer from 1 to 12
# Input: year of birthday, an integer from 1922 to 2010
day = 5
month = 5
year = 1998

# Calculate age
if month > 5 or (month == 5 and day > 5) :
    age = 2022 - year - 1
else:
    age = 2022 - year

#print age
print ('On 5 May 2022 I am ', age, ' years old')
```


Question 5

(a)

Questions

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Personal Identifier Z3537253

Started on Monday, 18 Apr 2022, 19:46

State Finished

Completed on Monday, 25 Apr 2022, 12:24

Time taken 6 days 16 hours

Marks 11.43/21.00

Grade 9.80 out of 18.00 (54.44%)

Information | Flag question

Welcome to the Block 1 quiz!

What is the purpose of the block quizzes?

The block quizzes are **not** meant to judge you. They are **formative** - you can get some marks via the TMAs, but these are for **engagement** with the quiz, not whether you answered the questions correctly. A score will be displayed for each quiz

A clear screenshot showing good engagement with the quiz.

Please follow the instructions in the assessment guide for your screenshot. It should be cropped to show only the relevant information (outlined in blue above) and should be taken before submitting the quiz. The mark is not important, but engagement is. Taking the screenshot before submission will indicate the level of engagement, even if the Python questions have not been attempted.

(b) The Quiz in the python section was challenging, I couldn't get it right, no matter if the program was getting the right result (output), I was still getting errors, it has to follow a restricted syntax. I don't like these integrated text editors, even if you copy and paste from the python IDE to the quiz It will still get errors, of course, you need to be careful copying and pasting code in python, since the indentation matters.

But in general, the rest was quite easy. And all the questions are related to week 1 and week 2 respectively.

(99 words)

It's good to see you engaging with the quiz and reflecting on how things are going. It would be good to explain what you learnt. Do make use of the forums; asking questions and contributing where you can, will help you to learn.

Note: Take care to read the instructions and follow them exactly. You must produce the exact output specified for the Python to be correct. For example, in question 18, 'Write a program that computes and prints the solution to the question "How many photos of 13MB each will fit in a 1000MB disk?"', you must format your answer exactly as specified: **76 photos of 13 MB fit in 1000 MB**

