

This is the practice Quiz for MATH2504. It is in a format similar to the actual quiz. More practice questions are in Practical D.

Instructions (for actual quiz):

- Write the quiz solution on blank paper, numbering each of the questions and ideally answering the questions in order. Make sure your name and student number are clearly written at the top of the paper.
- When finished with the quiz. Take photos/scans (using your phone is fine) and format the photos in a single document (e.g. paste in a Word document, or anything else). Then create a PDF file not bigger than 10MB and upload to BB using the quiz submission link.
- As with all MATH2504 assessment, create a short voice recording indicating the work is your own and stating that you followed all of the guidelines (if this is the case). Upload the voice recording as an additional file.
- The duration of the quiz is 50 minutes + 10 minutes reading time + 20 more minutes allocating to formatting the solution and creating the voice recording. Thus 80 minutes in total.
- Join the course Zoom link at the start of the quiz (actual quiz is Aug 31, 6:00pm BNE time). A link to the actual quiz will be provided. You can then ask questions (only during) the first 10 minutes (reading time) via private chat on Zoom. It is recommended you stay on Zoom for the duration of the quiz (until final upload at 7:20pm). This is in case there are announcements or comments. There is no need for you to turn on your camera or mic.
- You are not allowed to communicate with **anyone** during the quiz. The only communication allowed is asking questions to the instructor via Zoom private chat during the first 10 minutes.
- You are not allowed to run Julia or any other computational software during the quiz. You may use a hand calculator (or calculator on your phone), but not more.
- You are not allowed to use any material except for the course material directly from the course website (Units 1–3, Practicals A–D, HW1–HW2). That is, feel free to use your web-browser to look at the course materials during the practice quiz, however you are not allowed to use **any** other written material or any other material from the web.
- In your voice recording, if it is the case, you should clearly state that you didn't communicate with anyone, that you didn't run Julia or any other software, and that you didn't use any other material except for (perhaps) material from the course website.
- In case of exceptional circumstances (mishap with upload etc...), write the course coordinator. Otherwise, late submissions will not be accepted.

Quiz questions on next page...

Question 1: Consider the Julia code:

```
1: s = sum([i^2 for i in 1:100])
2: println(s)
```

1a: Replace line 1 above with lines 1–4 below to compute **s** using a **for** for loop (fill in lines 2 and 3 which will produce the same output):

```
1: s = 0
2:
3:
4: end
5: println(s)
```

1b: Do this now using a **while** loop by filling in exactly one line of code in line 3:

```
1: s, i = 0, 1
2: while true
3:     s += i^2
3:
4: end
5: println(s)
```

1c: Do this now using recursive function call by filling in code in line 5:

```
1: function f(s,i)
2:     if i == 0
3:         return s, 0
4:     else
5:
6:     end
7: end
8: s, _ = f(0,100)
9: println(s)
```

Question 2: Assume we wish to store numbers in an obscure way to only supports the values,

0, 1, 2, 3, 4, 5,
10, 11, 12, 13, 14, 15,
100, 101, 102, 103, 104, 105,
1000, 1001, 1002, 1003, 1004, 1005,
10000, 10001, 10002, 10003, 10004, 10005,
...
... and so on.

You store the numbers using a `UInt8` variable (1 byte) where the 3 first (right most/least significant) bits are used to represent the values 0, 1, 2, 3, 4, 5 in binary (call this value R) and the remaining 5 bits allow to represent the exponent in 10^n where $n = 0, 1, \dots, 2^5 - 1$. The number value which is stored is then $10^n + R$.

The following function returns a `UInt128` variable representing a number based on input x in the above form.

```
1: function to_value(x::UInt8)
2:     n = x >> 3
3:     R = x & 0b111
4:     @assert R<=5
4:     (n == 0 ? 0 : UInt128(10)^n) + R
5: end
```

So for example `println(to_value(UInt8(0b00011011)))` prints 1003.

2a: What output (or error) would you expect with the input `0xFF`?

2b: What is the largest input (in hexadecimal) that does not produce an error?

2c: Suggest a way to replace line 3 with an expression that uses only the `<<` and `>>` operators.

Question 3: Say that for some reason you wish to create a function that checks if an array is sorted using a divide and conquer strategy. The function needs to return `true` if the input array is sorted and `false` otherwise. You implement it as follows:

```
1: function is_sort(a)
2:     n = length(a)
3:     n == 1 && return ???
4:     m = n ÷ 2
5:     return is_sort(a[1:m]) && is_sort(a[(m+1):n]) && a[m] <= a[m+1]
6: end
```

3a: Determine code to replace the “???” in line 3 so that the function operates correctly.

3b: Say you apply the function to an array of 6 elements via `is_sort([1,2,4,3,10,7])`. Determine the total number of times `is_sort()` is called.

3c: Can you suggest a simple modification to line 5 which will reduce the number of times the function is called, at least for some inputs?