Your name: _____

Your student ID: _____

Instructions:

- This is an in-class quiz.
- Make sure to write your name and student ID above.
- Bring some form of ID card to the quiz. A student card is accepted. You will be required to show your id as you hand in the quiz.
- Answer the questions with pen or pencil on these sheets in class.
- Write answers to the questions exactly on the locations provided on the quiz sheets.
- You may use a calculator, but you cannot use a laptop, cellular device, or tablet. No other written material is allowed.
- Your answers should be short and without explanation unless one is explicitly requested.
- The duration of the quiz is 50 minutes + 10 minutes reading time.
- You are not allowed to communicate with anyone during the quiz and you are not allowed to look at the quiz paper of anyone else.. The only communication allowed is asking questions to the instructor at the end of the first 10 minutes as instructed. Any questions asked at that time will be answered to the whole group.
- In case of exceptional circumstances contact the course coordinator for special arrangements.

There are 9 items in total with each item worth 12 points.
This yields a total of 108 points total. The maximal grade is 100.

Quiz questions and on next page...

Question 1:

Consider the function `merge_bits`:

```
1:   function merge_bits(x1::UInt16, x2::UInt16)
2:       ans = 0x0000
3:       mask = 0x0001
4:       for i in 1:4532
5:           ans = ans | (x1 & mask)
6:           mask = mask << 1
7:           ans = ans | (x2 & mask)
8:           mask = mask << 1
9:       end
10:      return ans
11:  end
```

It takes two inputs of 16 bits, `x1` and `x2`, and returns a 16 bit output where the first output bit is the first bit from `x1`, the second output bit is the second bit from `x2`, the third output bit is the third bit from `x1`, the fourth output bit is the fourth bit from `x2`, and so on until the last output bit is the last bit from `x2`.

The above implementation of `merge_bits` does what is intended (it works).

1a: While the function works as intended a typo (or mistake) on line 4 implies that the loop does way too many iterations, many of which are unneeded. Replace `4532` with the smallest number possible so that the function still works as intended. What would you write instead of `4532`?

Your solution to 1a:

1b: What is the result of the expression `merge_bits(0x00ab, 0x00cd)`? Write your result in the format `0x____` (4 digit hexadecimal number).

Your solution to 1b:

1c: Another implementation of the function in only one line is in the form

```
merge_bits(x1::UInt16, x2::UInt16) = (x1 & 0xXXXX) | (x2 & 0xYYYY)
```

where `XXXX` and `YYYY` are each 4-digit hexadecimal numbers. Determine the values of `0xXXXX` and `0xYYYY` that would make this shorter implementation work.

Your solution to 1c:

Question 2:

Recall that the binomial coefficient, denoted as $\binom{n}{k}$, is defined as:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

where $n!$ denotes the factorial of n. Pascal's formula, which relates the binomial coefficients of different values, is given by:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$

with the base cases:

$$\binom{n}{0} = \binom{n}{n} = 1.$$

We also remind that $\sum_{k=0}^{n} \binom{n}{k} = 2^n$.

We also remind that $\sum_{n=0}^{N} 2^n = 2^{N+1} - 1$ (e.g. with $N = 3$ we have $2^0 + 2^1 + 2^2 + 2^3 = 2^4 - 1 = 15$).

Below is a recursive function that attempts to compute the binomial coefficient:

```
1:  function bin_coef(n::Int, k::Int)
2:      k > n && error("k must not be greater than n")
3:      (n == 0 || k == n) && return 1
4:      return bin_coef(n - 1, k - 1) + bin_coef(n - 1, k)
5:  end
```

2a: There is a typo on line 3 that causes a logical error with the function. What is the typo and the fix? Just suggest a simple character replacement.

Your solution to 2a:

2b: Assume that you fixed `bin_coef` correctly. Determine the value of the expression:

```
sum([sum([bin_coef(n,k) for k in 0:n]) for n in 0:10])
```

Briefly explain your answer.

Your solution to 2b:

2c: Assume that you fixed `bin_coef` correctly. Assume further that you replaced all of line 2 with `println("hey")`. How many times is `hey` printed when you evaluate `bin_coef(3,2)`?

Your solution to 2c:

Question 3:

A matrix of integer values is said to be monotonic if the entry at index (i, j) is not greater than the entry at index $(i + 1, j)$ and also not greater than the entry at index $(i, j + 1)$. With this description, if an index is out of bounds then the condition is ignored.

The following function is supposed to check if a matrix is monotonic and returns `true` if the matrix is monotonic and otherwise returns `false`. However there are two lines in the function that are wrong and each require a change.

```
1:   function is_monotonic(A::Matrix{Int64})
2:       m, n = size(A)
3:       for i in 1:m
4:           for j in 1:n
5:               cond1 = i < m && A[i,j] > A[i+1,j]
6:               cond2 = j < n && A[i,j] > A[i,j+1]
7:               (cond1 || cond2) && (return true)
8:           end
9:       end
10:     return false
11:  end
```

3a: Determine what changes to make on the function in two specific lines.

Your solution to 3a:

3b: Assume you fixed the function correctly. Consider the code:

```
A = [1 2 3 4;
     5 0 8 12;
     6 8 10 13]
println(is_monotonic(A))
```

What is the output, `true` or `false`? Also suggest a change of a single entry of the matrix that would change the output.

Your solution to 3b:

3c: Consider the code:

```
A = [1,2,3]*[10,20,20]'
is_monotonic(A)
```

What is the output, `true` or `false`? Briefly explain your answer (one or two sentences).

Your solution to 3c:

Do not forget to write your name and student number on the front page.