Mr. de Jonge wants to create an app for tracking exposure to people infected with the disease CO-20. in The Netherlands. The idea is that phones exchange numbers with each other. If a person becomes infected, the numbers emitted by that persons phone are distributed to all phones. If any of the infected numbers was recently received by a phone, the app will issue a warning that its owner might be at risk of having been infected. *(*)*

Details:

- The app generates a new number 64 times per day.
- There are 16 million people in the Netherlands.
- Every number should be unique for 14 days.
- The number must be represented as a sequence of max 8 characters.

You are asked for advice on the data representation of these numbers.

1. How many bits are required at minimum to guarantee the uniqueness of numbers?
2. Which number representation should be used?
3. How much data is generated at maximum after a day? Round to whole GiB.

Explain your answers.

(*) Please note that any resemblance to actual persons, living or dead, or actual events is purely coincidental.

We are going to implement CO-hash. The idea is that we are going to change your student number such that you do not have to remember 7 digits anymore, but something "simpler" to remember that will (most likely) still be unique for at least your cohort! So what you need to do is the following:

1. take your student number (hover over your name in the top left corner)
2. take the last 2 digits modulo 36; this is your *base*
   2.1 if that results in `00`, `01`, `02`, `08`, `10`, or `16` add `13`. (So change to `13`, `14`, `15`, `21`, `23`, or `29` respectively)
3. Convert your student number to this new base.

Make sure to write the whole calculation.

Ms. ‘Sweet sixteen’ has designed her own floating point standard, which starts off with a sign bit, followed by a 5-bit exponent (in excess-32), and a 10 bit mantissa with a ‘hidden’ bit just like big brother IEEE 754.

Consider the following number:

```
0 11001 0111010000
```

Derive the resulting number (bit string) when multiplying it with the scalar

```
6
```

Hint: multiplication can be expressed in terms of shifts and additions. **Show your work!**

Romeo Duck and Juliet Shark got into a bit of a fight. Romeo Duck says that if you use BCD you cannot just add these numbers together as the result will not be BCD anymore. Juliet Shark on the other hand says that it can be done, but you just need to add 6 to the resulting BCD digit if there is overflow (e.g.: if the addition of 2 BCD digits results in `1111` (15); add `0110` (6), which results in `0001 0101`, or 15).

Given the numbers 36 and 75, your task is to preserve love and peace in the animal kingdom and prove them both right.

1. Prove Romeo Duck right by showing that regular binary addition of these two numbers does not result in a valid BCD number.
2. Prove Juliet Shark right by showing that her solution works to get the correct result in BCD.

We are designing a new ISA. This new laptop ISA will be specifically targeting the needs of the students. This means the ISA needs to be efficient at streaming video and copying bytes. What has been decided so far is the following:

- 42 bit instructions
- minimum 4.2; maximum 42 GiB of RAM
- 42 registers of 256 bits
- byte addressable

What problem do you see with this ISA and how would you solve that?

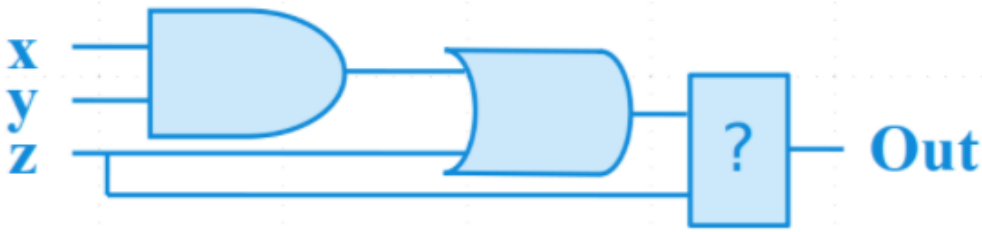
While working on the Self-study assignments, your teammate accidentally spills coffee over your Karnaugh-map. It has become completely unreadable and you will have to redo the whole map to determine the minimal sum-of-products.

The formula that you were trying to minimize is
$ACD + C\bar{D} + \bar{A}\bar{B} + BCD$

1. Fill out the Karnaugh-map below.
2. Determine the minimal sum-of-products.
3. Explain in max two sentences how you can get a minimal sum-of-products formula from a Karnaugh map.

		AB			
		00	01	11	10
CD	00				
	01				
	11				
	10				

Copy the following into your answer and fill in the blanks.



Which gate needs to be inserted at the ?-box in order to get the desired output as listed in the truth table?

X	Y	Z	Out
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Select one answer

- [ ] AND
- [ ] OR
- [ ] XOR
- [ ] NAND
- [ ] NOR

Ducks decided some 5 years ago that they need to start a census to determine once and for all whether there are more sharks or more ducks in the world. They discovered that in those 5 years the number of ducks doubled every year. They store the amount of ducks in CO-42 which has the following properties:

- 2 sign bits
- 8 exponent bits, excess-127 (same rules and special numbers as IEEE-754)
- 32 mantissa bits

There were 65536 ducks when they started counting. In what year will the amount of ducks no longer fit?

You are given the following piece of `Assembly x86-64` code:

```
1   popq    %rdi
2   popq    %rsi
3   call    foo
4   incq    %rax
5   pushq   %rax
6
7   movq    $0, %rdi
8   call    exit
9
10  foo:
11      pushq   %rbp
12      movq    %rsp, %rbp
13
14      addq    %rsi, %rdi
15      pushq   %rdi
16      pushq   %rdi
17      movq    %rsi, %rax
18
19      movq    %rbp, %rsp
20      popq    %rbp
21      ret
```

Before `line 1` is executed, the 8-byte aligned stack looks like this:

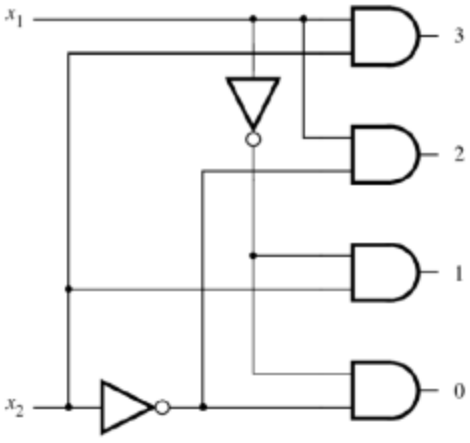
Address	Contents	
464	"COCONUT="	RBP
456	"=NOTANUT"	
448	20	
440	22	RSP
432	0	
424	837	
416	351	
408	656	

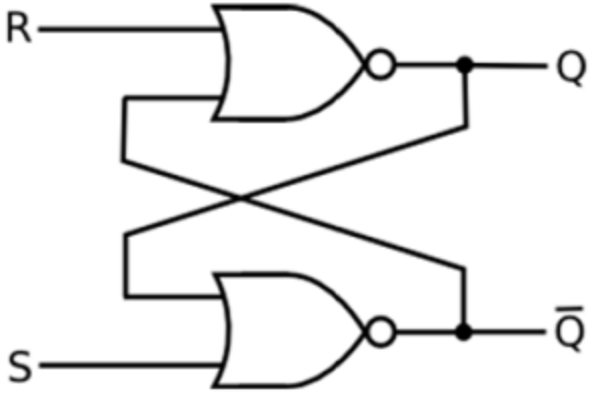
1. What will the stack look like after `line 17` is executed?
2. What will the stack look like after `line 5` is executed?

You're applying for a software developer position at the tech giant company **Ducks & co.** and you got invited to an interview. In the invite you were asked to prepare an answer to the following problem:

A design for a digital circuit is needed which adds three input bits: a, b, and c. The output of this circuit should be the carry bit C. What formula can be used to create this circuit? Show how you got to the formula.

Below are shown two digital circuits. These are:





Select one answer

- [ ] Mulitplexer and D flip-flop
- [ ] Decoder and SR latch
- [ ] Multiplexer and SR latch
- [ ] Decoder and D flip-flop

Which of the following equations **are** equal to each other?

Select all that apply

- [ ] $ca + a\overline{b} + cbd + \overline{c}\overline{bd} + cadb$
- [ ] $ca + da\overline{b} + cbd + \overline{c}\overline{d}\overline{a}\overline{b}$
- [ ] $a\overline{b} + \overline{c}\overline{bd}\overline{a} + cba + cbd$
- [ ] $ca + cbd + \overline{c}\overline{b}a + \overline{c}\overline{bd}$

Select all that apply

- [ ] the difference between D latch and D flip-flop is that D flip-flop does not need a clock

- [ ] a clock can be both rising and falling edge triggered

- [ ] A clock can be both edge and level triggered

- [ ] level triggered is preferred over edge triggered