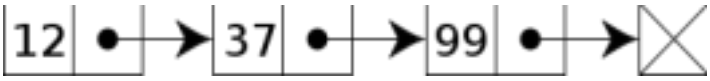
1. Algorithms and Data Structures

1.1. You were given a singly linked list of integers:



You need to return a list with the elements in the reverse order. Describe an algorithm, for the generic case of a list containing N elements that would do such work in a linear time complexity.

R: Assuming that I already have the linked list (if not, I will implement the class “Node” to get and set the values of the current and the next node), I can create the reverse function in this order:

- Set the previous value as None (null);
- Set the current value as the head of the linked list;
- Itherate the linked list and store the next value in a variable (until the end of the linked list);
- Set the next value of the current item as None (null);
- Set the previous value as the current item;
- Set the current item as the next; and
- After reaching the end of the linked list I set the head of the linked list as the previous item.

I have implemented the practical example in the file “scripts/exercise 1.1.py” (with comments).

1.2. You were given an array of unknown length containing sorted elements that if you access a random position out of the bounds of the array an exception is thrown. Describe an efficient algorithm to search an element.

R: In this case we have several approaches:

- Using python we can just check if the desired value is in the array;
- Using python without the built-in list check but with the bult-in functions, we can use the “len()” function to return the size of the array and then use a binary search to return the desired value; and
- Assuming we need to implement an “universal” solution, we need to start a search in the list that increments in a quadratic way, checking if the current value is lesser or greater than the desired valued and after the cut, we apply the binary search.

Obs.: The binary search will be used since all elements in the array are sorted.

All approaches will be presented in the practical example “scripts/exercise 1.2.py” (with comments).

2. Quality Assurance and Scripting

2.1 Imagine that you are responsible for guaranteeing the quality of a software that is constantly updated. How would you guarantee that those updates will not affect parts of the software that were working correctly before?

R: As the software is constantly updated, the first step to maintain and guarantee that the software works as intended is implementing automated tests. These tests should be run after the development process and if it fails the code must be reviewed and adjusted. The main focus of the automated test of this type of software is the regression test to ensure the quality of the current system.

This can be done with CI/CD pipelines to deploy if all tests results in success.

2.2 Consider that you have built a 100-floor building and that you have just asked a company to install an elevator. Write a list containing all tests that you would do to check if it works as intended (remember that this is a very tall building and there is only one elevator

R: I would consider checking:

- The elevator stops at all floors?
- The elevator stop spot at all floors is aligned?
- All the buttons works as intended?
- All alerts are working?
- The elevator will operate above the maximum weight limit? It isn't allowed.
- If the elevator cable breaks the emergency break will be actioned?
- During an electricity intermittence the elevator doors can be opened?
- During an electricity intermittence the elevator system have any backup generator?
- The elevator has an emergency exit (a small trap door)?
- During a fire, fire alarm or fire tests the elevator can be used / operated? (if don't have somebody inside)
- The elevator speed is sufficient to reach all floors in an acceptable time?
- The elevator provides any trustful communication system?
- The elevator provides communication to emergency systems?
- The elevator system is configured to attend the queue in the most efficient way?
- The elevator provides comfort and air conditioner system?

2.3 How many times do you execute a task until you decide to automate it? Explain why.

R: When I finish a task and I see that is possible to automate. If the task is possible to be automated but is more complex to do it I spend time studying the task, elaborating the requirements and then I start developing a simple solution that will be incremented over the time until I decides its "stable" or "finished". Simple tasks automations are finished after a few adjustments to the original project.

2.4 A certain system needs a password validator module, which upon receiving a string with a password and a list with the requirements of this password, return whether the password is valid or not. The list of the password requirements is composed of tuples containing the following:

• First value: o LEN – password length o LETTERS – # of letters o NUMBERS – # of numbers o SPECIALS – # of special characters

• Second value: <, > or =

• Third value: an integer number Ex.: req = [('LEN', '=', 8), ('SPECIALS', '>', 1)] req specify a password with eight characters and at least two special characters.

Write a Python 3 script to solve this problem and the unit test to validate it, without installing external packages.

R: This answer can be found in the file "scripts/exercise 2.4.py" (with comments). In the script, I'll perform all tests, demonstrate several errors that were supressed by the "try / except" clause and an error that raises an exception and terminates the script.

3. Hardware and Simulation

3.1 Draw the circuit that would be generated from the Verilog/VHDL description right below. Both descriptions are equivalent.

Verilog

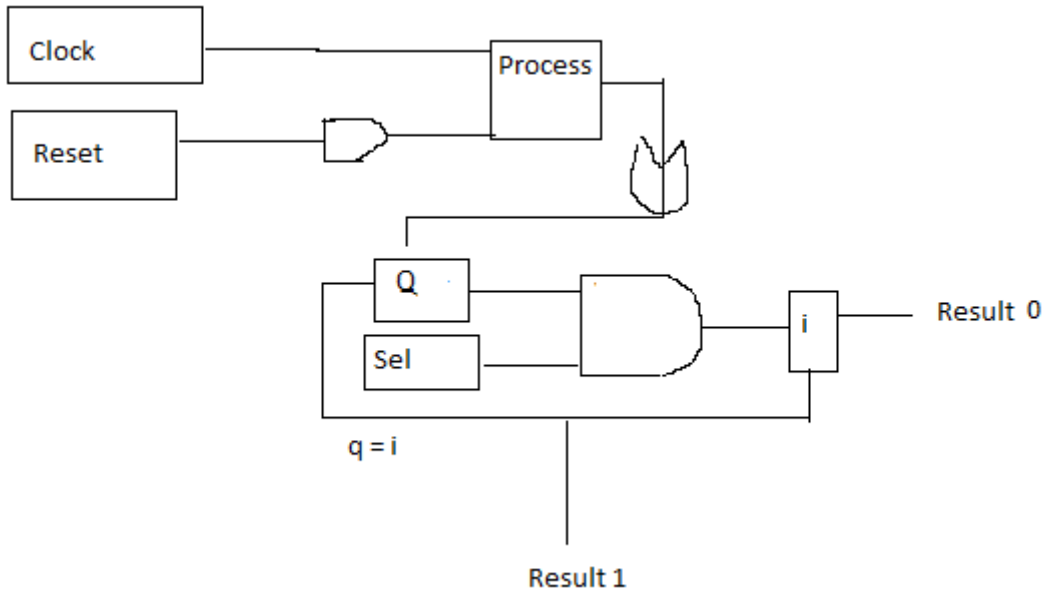
```
module test (clock, reset, sel,
result);   input clock, reset, sel;
output result;   reg [1:0] q;   wire
[1:0] i;
    assign i =
{q[0],sel};   always
@(posedge clock)
      if (reset) q = 2'b0;
      else q = i;
    assign result = (q == i) ? 1'b1 :
1'b0;

endmodule
```

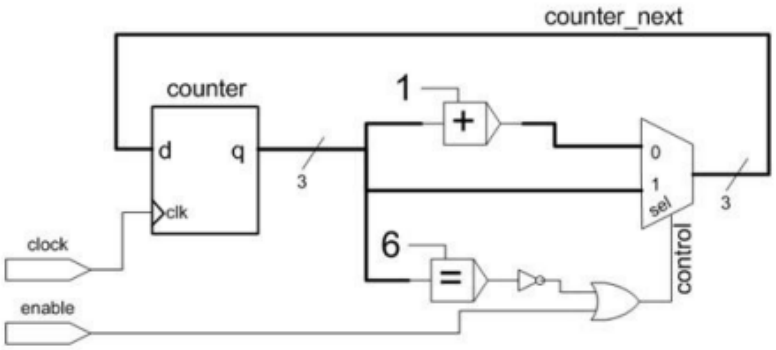
VHDL

```
library IEEE; use
IEEE.std_logic_1164.all;
  entity test is   port (
      clock,reset,sel: in std_logic;
      result: out std_logic
  ); end
entity;
  architecture rtl of test
is
  signal q,i: std_logic_vector(1 downto 0);
begin
  i <= q(0) & sel;
process (clock, reset)
begin
      if (rising_edge(clock)) then
      if (reset = '1') then      q
<= "00";          else
     q <= i;
      end if;
      end if;
  end process;   result <= '1' when
(q = i) else '0'; end architecture;
```

R:

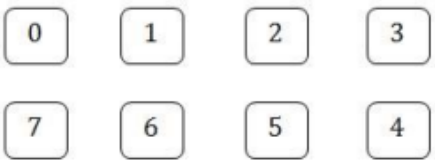


3.2 Consider the circuit below:

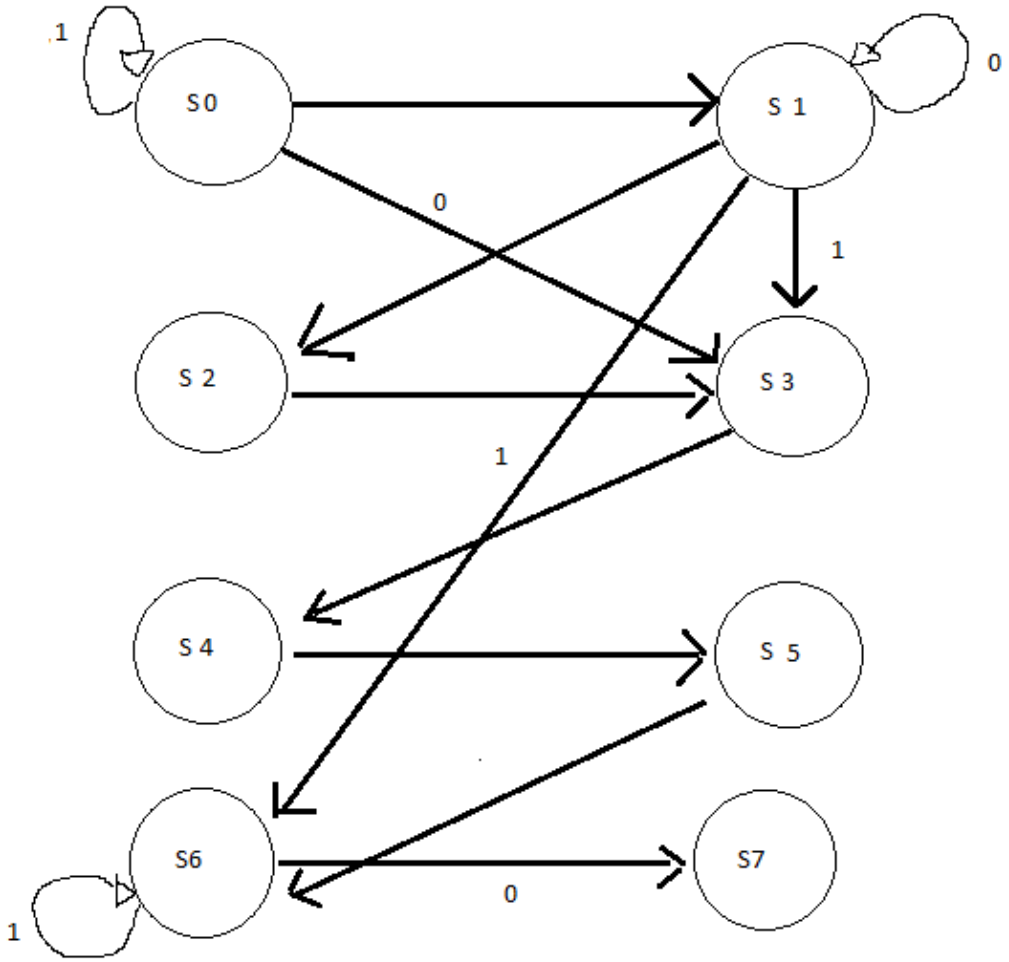


Complete the state transition diagram below for the register "counter". Consider that the finitestate machine makes a transition when "clock" has a positive edge and that every signal can only take the values 0 or 1 (no X values).

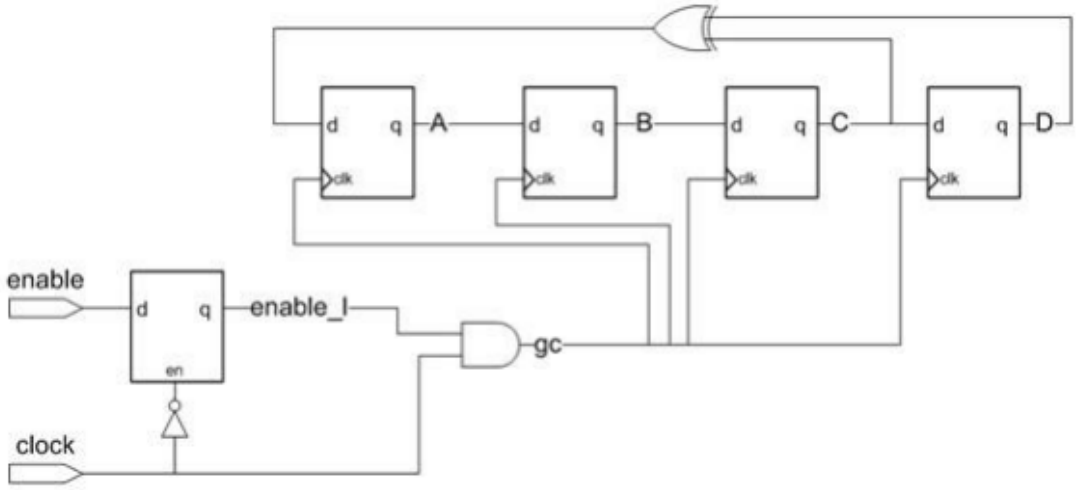
Tip: Don't assume anything about the circuit behavior based on signal names.



R:



3.3 Given the circuit below, complete the waveform:



Tip: The green-filled cycles are considered as an undefined value X, such as a memory element that has not been initialized or reseted.

R: