The link to the GitHub repository: <https://github.com/00009815/CSF.CW1.00009815/tree/exam>

**Task 2**

1. ID = 9815

9815/2 = 4907 Remainder: 1

4907/2 = 2453 Remainder: 1

2453/2 = 1226 Remainder: 1

1226/2 = 613 Remainder: 0

613/2 = 306 Remainder: 1

306/2 = 153 Remainder: 0

153/2 = 76 Remainder: 1

76/2 = 38 Remainder: 0

38/2 = 19 Remainder: 0

19/2 = 9 Remainder: 1

9/2 = 4 Remainder: 1

4/2 = 2 Remainder: 0

2/2 = 1 Remainder: 0

1/2 = 0 Remainder: 1

9815 in decimal = 10011001010111

Binary: 10 0110 0101 0111 => Hex: 2657

9815 in hex = 2657

1. Remainder

99999/2 = 49999 1

49999/2 = 24999 1

24999/2 = 12499 1

12499/2 = 6249 1

6249/2 = 3124 1

3124/2 = 1562 0

1562/2 = 781 0

781/2 = 390 1

390/2 = 195 0

195/2 = 97 1

97/2 = 48 1

48/2 = 24 0

24/2 = 12 0

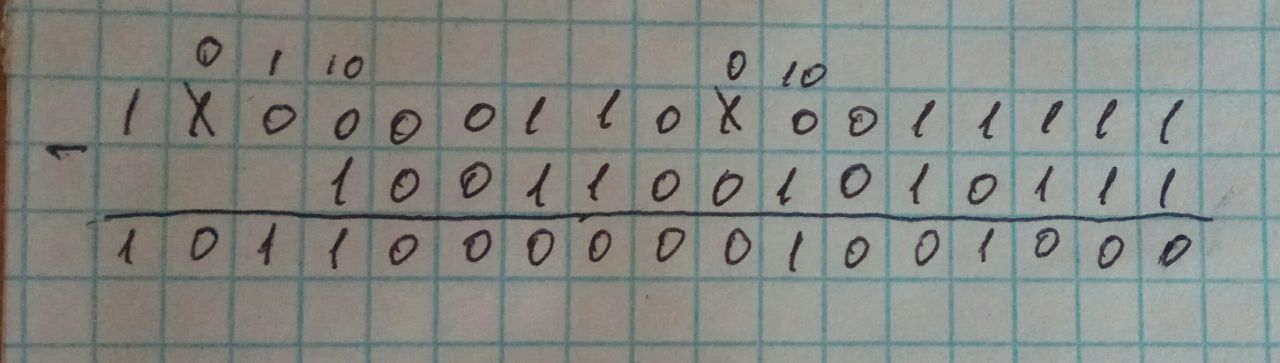
12/2 = 6 0

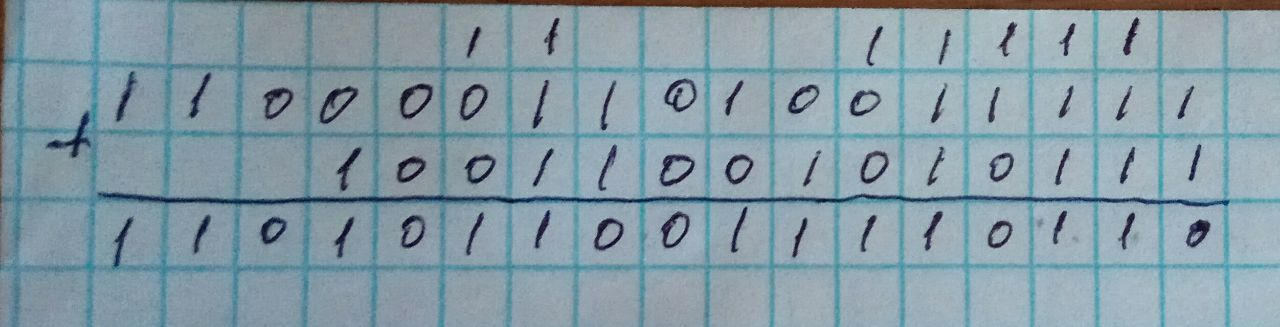
6/2 = 3 0

3/2 = 1 1

1/2 = 0 1

99999 in decimal = 11000011010011111



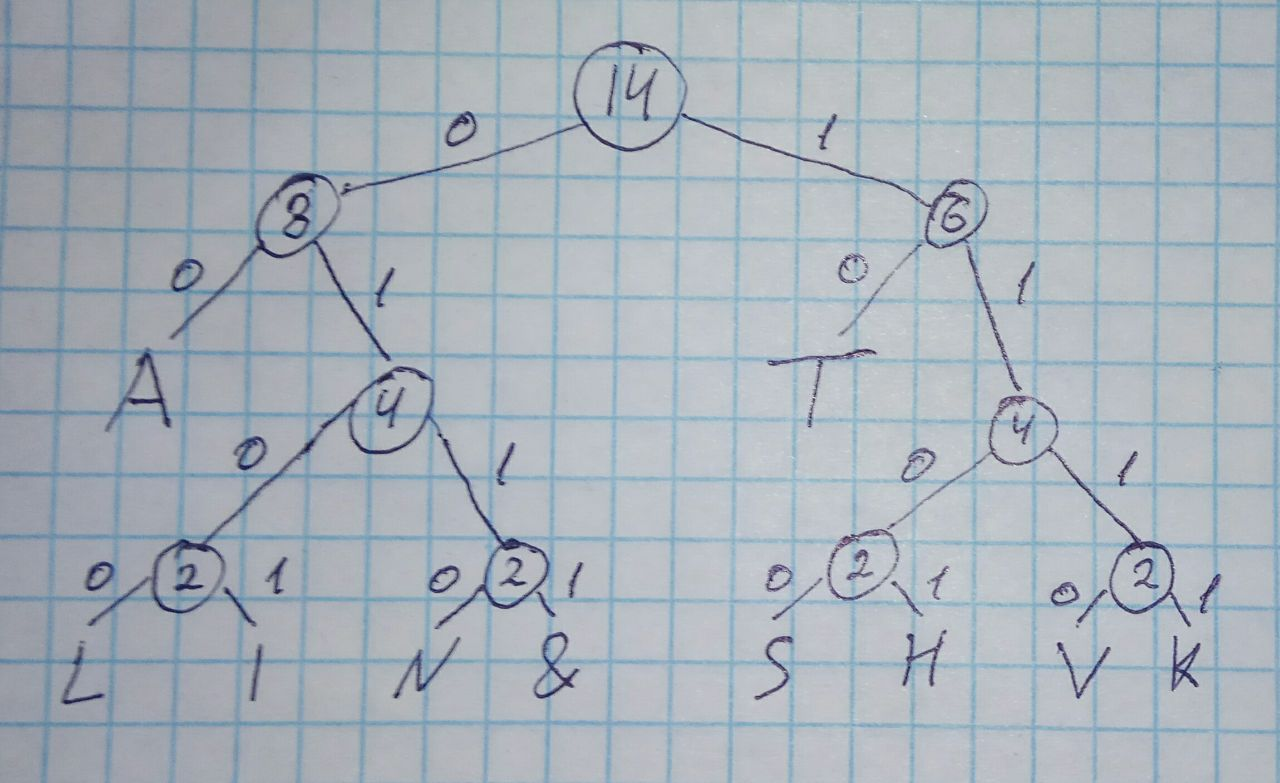


1. Hexadecimal system is more concise than binary and allows to represent large binary numbers in a few digits. Additionally, it is easy to convert between the hex system, decimal and binary. Hexadecimal is used in definitions of memory locations, colours, MAC addresses and error messages (Teach Computer Science, 2018).

**Task 3**

Altina&Shavkat

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | L | T | I | N | & | S | H | V | K |
| 4 | 1 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |



Altina&Shavkat => 00010010010101100001111100110100111011110010

Total number of bits = 4\*2 + 1\*4 + 2\*2 + 1\*4 + 1\*4 + 1\*4 + 1\*4 + 1\*4 + 1\*4 + 1\*4 = 44

**Task 4**

The list of numbers: 1 2 3 4 5 7 8 9

The wanted number: 8

Pseudocode:

X = 8 (wanted number)

lowerHalf = 1

upperHalf = array\_length

while x not found

midpoint = lowerHalf + (upperHalf – lowerHalf) / 2

if midpoint < x

lowerHalf = midpoint + 1

if midpoint > x

upperHalf = midpoint -1

if midpoint = x

x found

if upperHalf < lowerHalf

x is absent

1. Midpoint = 4 (8/2)
2. X > midpoint (8>4), drop the left side
3. Midpoint = 7 (4/2)
4. X > midpoint (8>7), drop the left side
5. Midpoint = 8 (2/2)
6. X = midpoint (8=8), number found

**Task 5**

In paged memory technique all processes are separated into pages and stored in memory frames. The frame is a part of memory allocated for a process page. Page is a part of a process in a memory frame. A frame and a page must be the size (Griffin, 2019).

1. <2,85> 5\*1024+85 = 5205
2. <0, 1026> Invalid address. The offset is bigger than the frame size.

**Task 6**

Waterfall:

* Divided into separate stages of the development
* Demands thorough planning in the beginning
* The development is not flexible

Suitable for simple projects that does not require constant customer feedback (Waddell, 2019).

Agile:

* The project is divided into parts, which are completed in a number of sprints
* Review and feedback after each sprint
* The development can adjust to the changing customer requirements

Suitable for complex projects that require early working builds and customer feedback (Trapani, 2018).

**Task 7**

Bus topology – all devices are connected to the same cable.

Advantages:

* Easy to install
* The usage of the cables is lower, comparing to the other topologies
* Cheap installation due to the small number of cables required

Disadvantages:

* Challenging to locate the problem in the network
* The whole network stops to work if the main cable is damaged
* The speed decreases as the number of devices increases

Suitable for small networks with a limited number of devices connected (Computerhope, 2018b).

Ring topology – each device is connected to the two neighboring devices, forming a circle.

Advantages:

* The number of connected devices does not affect the network performance, as the data only flows in one direction.
* No server is required to manage the device connection

Disadvantages:

* Data must pass through connected devices which lowers the speed
* If one device switches off, the whole network stops working

Suitable for small and medium size networks (Computerhope, 2018a).

**Task 8**

# void function

# it does not return any value and has no effect on the flow of the code

def said(person, speech):

print(f"{person} said {speech}")

# value returning function

# has 'return' keyword. Returns value in the script and directly affects the performance of the program

def squared(number):

return number\*number

said("Cato", "Carthago delenda est")

Answer = 56 + squared(32) - (17 + squared(12))

print(Answer)

**Task 9**

# object-oriented approach is more concise and allows to reuse the same code multiple times

# class that creates an instance of a client with one's name and work duration. Has a function to calculate fee

class Client:

def \_\_init\_\_(self, name, days):

self.name = name

self.duration = days

def fee(self, type):

if type == 1:

return self.duration \* 100

elif type == 2:

return self.duration \* 200

elif type == 3:

return self.duration \* 300

else:

return print("Invalid input")

cl1 = Client("Cato", 30)

print(cl1.fee(2))

# functional approach requires writing separate functions for different portions of the program

# function to calculate client's fee, taking information from the client object

def fee\_counter(client):

if client["type"] == 1:

return client["duration"] \* 100

elif client["type"] == 2:

return client["duration"] \* 200

elif client["type"] == 3:

return client["duration"] \* 300

# client object

cl2 = {

"name": "Scipio",

"duration": 45,

"type": 3

}

print(fee\_counter(cl2))

Reference list

Computerhope. (2018a). What is a Ring Topology? *Computerhope.com*. Available from https://www.computerhope.com/jargon/r/ringtopo.htm [Accessed 14 January 2021].

Computerhope. (2018b). What is Bus Topology? *Computerhope.com*. Available from https://www.computerhope.com/jargon/b/bustopol.htm [Accessed 14 January 2021].

Griffin, L. (2019). Paged Memory Allocation: Definition, Purpose & Structure - Video & Lesson Transcript. *Study.com*. Available from https://study.com/academy/lesson/paged-memory-allocation-definition-purpose-structure.html [Accessed 14 January 2021].

Teach Computer Science. (2018). Uses of Hexadecimal | Hexadecimal & Character Sets | Computer Science. *Teach Computer Science*. Available from https://teachcomputerscience.com/uses-of-hexadecimal/.

Trapani, K. (2018). What is Agile/Scrum. *cPrime*. Available from https://www.cprime.com/resources/what-is-agile-what-is-scrum/ [Accessed 14 January 2021].

Waddell, J. (2019). The Cascading Costs of Waterfall. *Medium*. Available from https://medium.com/@joneswaddell/the-cascading-costs-of-waterfall-5c3b1b8beaec [Accessed 14 January 2021].

‌