**Task 2**

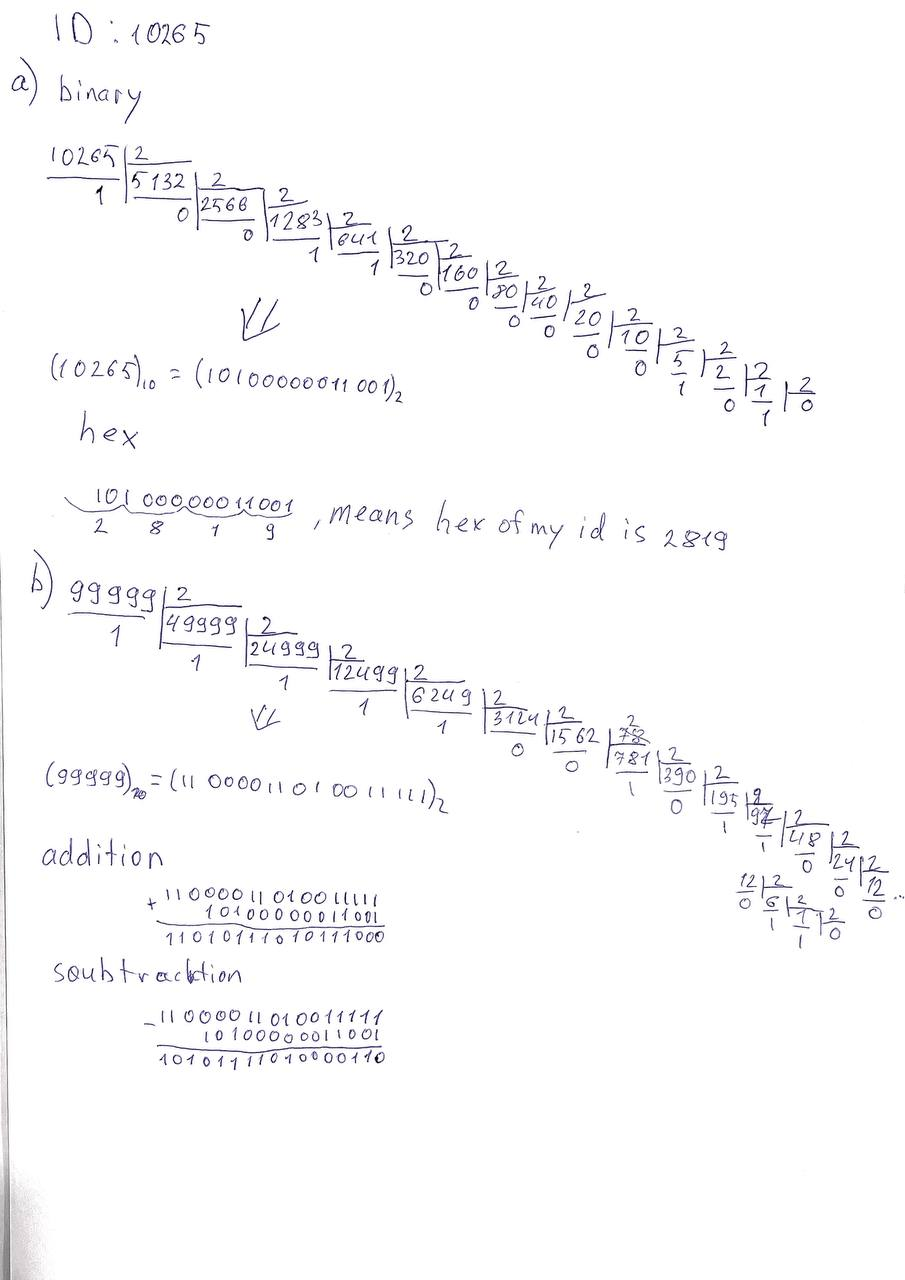
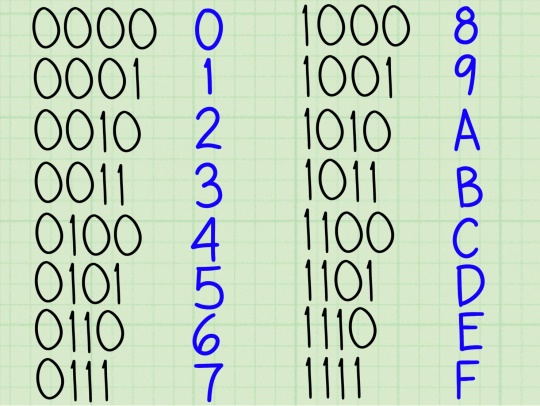
1. My ID is 10265

Binary of my ID is

Using hex-binary table we can find that hex of my id is 2819

Addition => + =

Subtraction => - =



1. Hexadecimal numbers provide better opportunity to use binary numbers in more compact way and therefore are used in Assembly code. In Addition, they represent colors used in HTML and CSS or various symbols in ASCII. They are also used in MAC (Memory Access Control), which is a unique number of a device in the Internet.

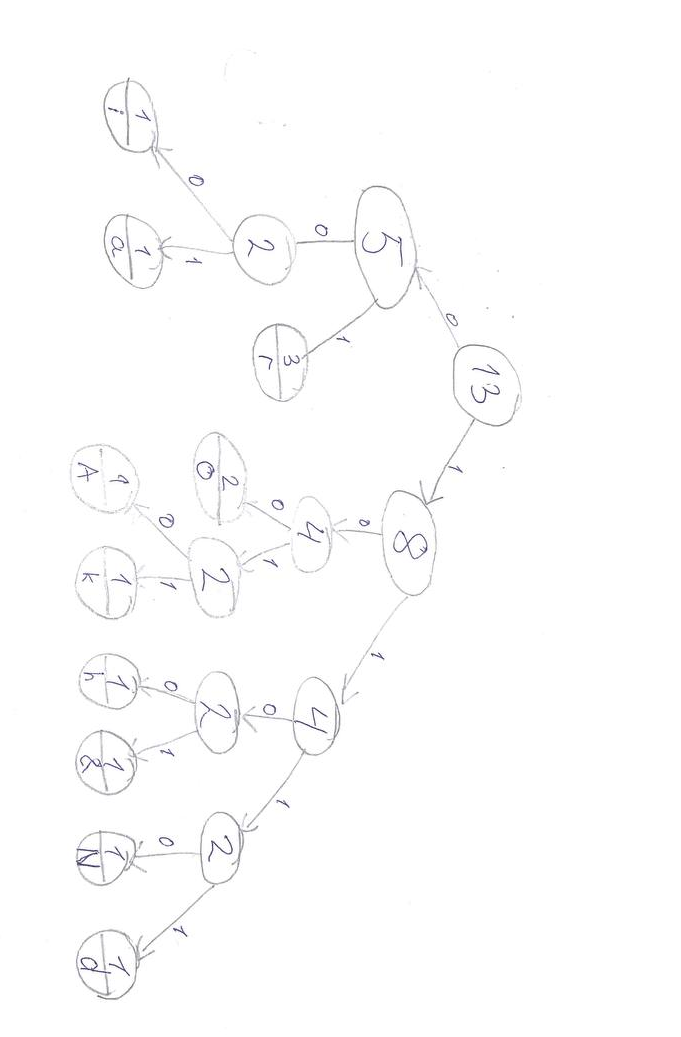
**Task 3**

Father’s name- Akhror

Mother’s name – Nodira

Sentence – Akhror&Nodira

|  |  |  |  |
| --- | --- | --- | --- |
| Symbols | Frequency | Code | Code length |
| r | 3 | 01 | 2 |
| o | 2 | 100 | 3 |
| i | 1 | 000 | 3 |
| a | 1 | 001 | 3 |
| A | 1 | 1010 | 4 |
| k | 1 | 1011 | 4 |
| h | 1 | 1100 | 4 |
| & | 1 | 1101 | 4 |
| N | 1 | 1110 | 4 |
| d | 1 | 1111 | 4 |



The code is 101010111100011000111011110100111100001001

Code length is sum of frequencies times code lengths = 3\*2+2\*3+3+3+4\*6 = 42 bits

**Task 4**

1,0,2,6,5,4,5,2,3,7 => length is 10

Array = 1,0,2,6,5,4,5,2,3,7

1. Sorting the list

REPEAT

Swapped = False

FOR i=0 to length(Array)

IF Array(i)>Array(i+1) THEN

X= Array(i)

Array(i)=Array(i+1)

Arry(i+1)=X

Swapped = True

END IF

NEXT i

UNTIL Swapped = False

Sorted array is = {0,1,2,2,3,4,5,6,7}

Now the pseudocode for the binary search

Highest= HighestBoundary (Array)

Lowest= LowerBoundary (Array)

Do WHILE Lowest <= Highest

Middle= (Lowest+Highest)/2

IF num= Array(Middle) THEN

Found = True

EXIT DO

ELSEIF num< Array(Middle) THEN

Highest=(Middle-1)

ELSE

Lowest = (Missle+1)

END IF

LOOP

If we want to find 6, for example, we go through all this process.

1. The midpoint is 3
2. 6 is larger the 3 therefore left part is gotten rid of and new Lowest is 4
3. Midpoint is 5.5
4. New Lowest is 6
5. Midpoint 6.5
6. New highest is 6
7. Number found

**Task 5**

We have covered three types of memory management techniques: Single Contiguous, Partition Memory, and Paged Memory Management.

Paged Memory Management fractionates memory into parts that are called frames, which can be used by any programs in any order. Unlike this type, Single Contiguous memory management allows division of memory only between OS and application program. This wastes a lot of memory, as the program usually does not need that much memory. Partition memory, meanwhile, also divides memory into parts and allocates them to the programs. However, one advantage of the Paged Memory technique is that contiguous storage in memory is not required. This means, because the process is divided into parts, it is easier to load them rather than loading one large process.

Logical address described as <page, offset>

Physical address = frame\*frame size + offset. Frame size = 1024

1. Frame 2 means page=5

Physical address = 5\*1024+85=5205

1. Invalid offset. It may not be larger than the frame

**Task 6**

Waterfall - the traditional project management methodology – is a linear approach. It usually has five phases- gather requirements, design, code and unit test, system testing , deliver the finished product; a new phase begins only after the previous phase completed.

Agile, in contrast, is not linear, but iterative. Instead of creating plan for the whole project, it divides development work into small increments completed in iterations.

Here are the main differences between the two:

Waterfall – a linear and sequential

Waterfall divides a project into phases

Waterfall is good for one single project

Waterfall aims successful project delivery

Waterfall requires a project manager

Waterfall avoids scope changes after the project has started

Waterfall creates requirements at the start

Agile – incremental and iterative

Agile divides a project into sprints

Agile helps complete many small projects

Agile aims customer satisfaction

Agile enables entire team to manage the project

Agile may introduce changes at any time

Agile prepares requirements along the process

Waterfall methodology is not very popular among companies and mostly used by government organizations and federal agencies. Because it is more customer-oriented, Agile methodology would be more advisable for the companies. Also, it allows to shape your product based on the assessment of each iteration. Only problem may be the funding, because if there is fixed-price scenario, it may increase stress. Nevertheless, this methodology is still better where it is feasible.

**Task 7**

**Bus**

This type of network has one large backbone cable and all the computer and other devices (nodes) are connected to it. On both ends of the cable there is a terminator that absorbs signals. All the signals are sent from one node to another through main cable (which also may cause signal bounce – when signals from different nodes collide).

Advantages

* Requires less cable and therefore cheaper option
* It is easy to connect new nodes to the system

Disadvantages

* Problems with main cable may cause breakdown of the system
* It is also difficult to identify the problem

Mostly used for small, cheap or temporary network, that does not require large data transmissions. These may be laboratories, as well as small offices.

**Star**

This topology has a central device and all nodes are connected to it. Large network usually use this topology and have one server that controls it.

Advantages

* Because nodes are not connected to each other, failure of one does not go viral
* It is easy to connect new nodes to the system
* Better performance (due to lack of any signal bounces)

Disadvantages

* Problems with switch may cause breakdown of the entire system
* Requires a lot of cables
* Large amount of information passing through the switch may cause network slowdown

This type of network is used mostly by large organizations, rewiring high performance. Also, our Wi-Fi routers at home are also a clear example of star network.

**Task 8**

# To explain briefly, returning function returns a value, which can be used further, while void functions do not.  
# Let’s proceed to examples to understand this better:  
# returning function  
  
  
def get\_surname(name):  
 return name + 'a'  
# void function  
  
  
def print\_surname(name):  
 print(name+ 'a')  
  
  
# In the first case we need to add print function to receive the answer  
print(get\_surname('Yoqubov'))  
  
# In the second case we just call the function, because it does not return, but just prints it  
print\_surname('Yoqubov')  
  
# Further, in the following case we give a variable a value  
surname = get\_surname('Yoqubov')  
print(surname)  
  
# here we get ‘Yoqubova’  
  
Surname2 = print\_surname('Yoqubov')  
print(Surname2)  
  
# Here, however, we will get ‘none’ because the function did not return any value, so the variable  
# Thus, we can conclude that when we need the function to return the value that can be used further,  
# we use the first type, while we need the function to just do short use of the information without returning  
# we use the second option.

**Task9**

Functional programming is a programming paradigm in which we try to bind everything in pure mathematical functions style. OOP, meanwhile, is based on concept of the objects containing data.

import functools

# To see the difference between the two, I will show how to solve one task on both of them

# We should calculate sum of the following list

the\_list = [1, 2, 3, 4, 5]

# Functional programming aims to convert everything to math equation

def functional\_programming(x, y):

return x + y

sum\_list = functools.reduce(add\_it, the\_list)

print(sum\_list)

# Object oriented programming

class SumList(object):

def \_\_init\_\_(self, any\_list):

self.any\_list = any\_list

def do\_add(self):

self.sum = sum(self.any\_list)

sum\_of\_list = SumList(the\_list)

sum\_of\_list.do\_add()

print(sum\_of\_list.sum)

This short example clearly describes how these two work.

Thus, we can make an assumption that functional paradigm is better when we have fixed things to be operated with. This way we can use full potential of all functions.

In contrast, OOP is more adaptable for fixed sets of operations, and further add new items. In other words, we may create more classes to carry out operations with.

**Reference list**

B. (n.d.). Network topologies - Communication - Eduqas - GCSE Computer Science Revision - Eduqas - BBC Bitesize. Retrieved January 15, 2021, from https://www.bbc.co.uk/bitesize/guides/z7mxh39/revision/5

Dale, N., & Lewis, J. (2020). *Computer science illuminated*. Burlington, MA: Jones & Bartlett Learning.

Santos, J. (2020, October 06). Agile vs. Waterfall: Software Development Methodologies. Retrieved January 15, 2021, from https://project-management.com/agile-vs-waterfall/#:~:text=Agile%20is%20an%20incremental%20and,helps%20complete%20one%20single%20project.