Task 2

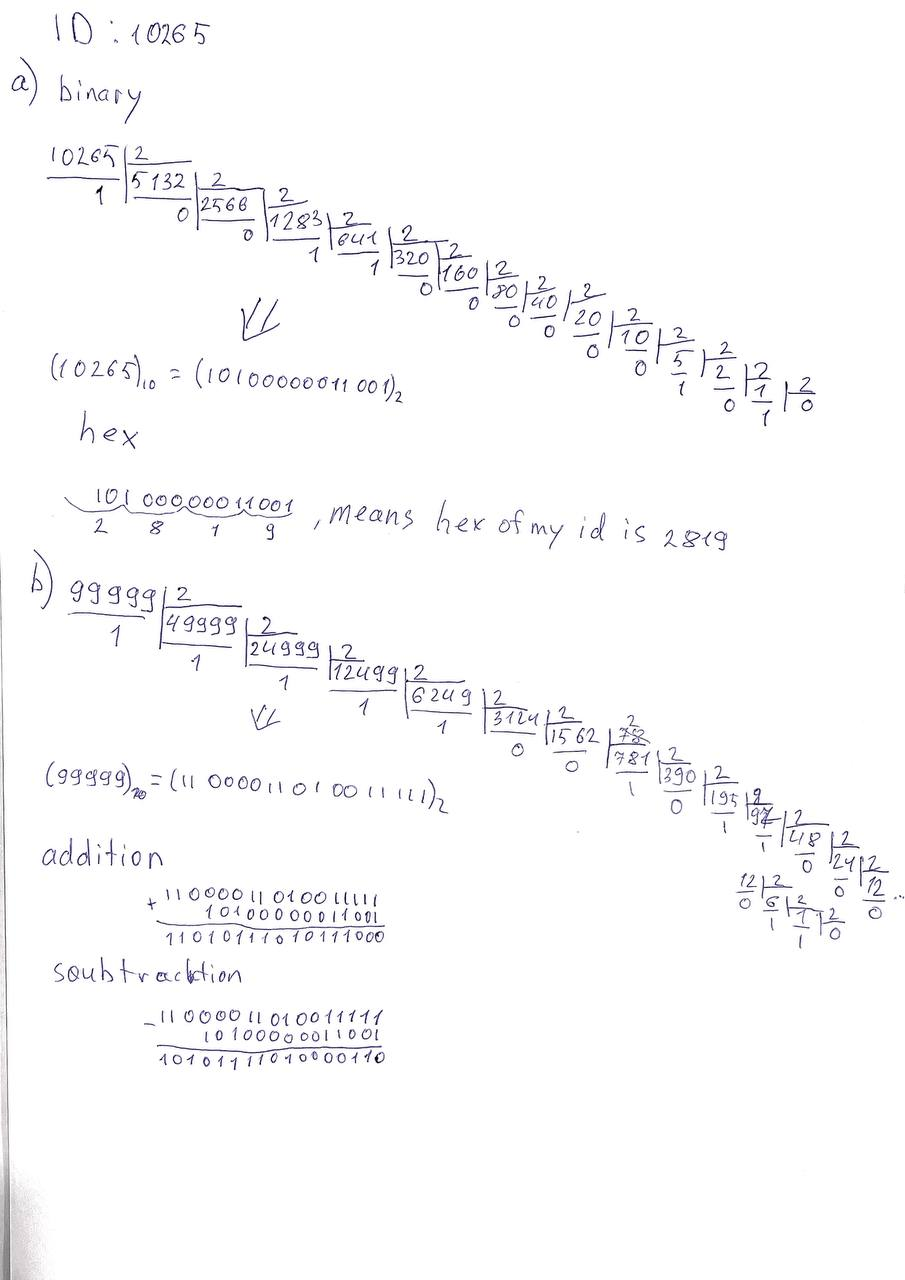
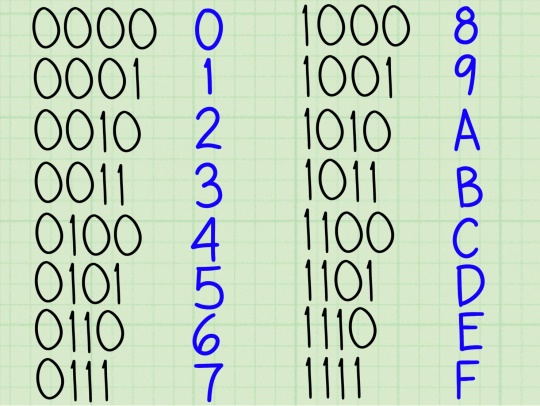
1. My ID is 10265

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.

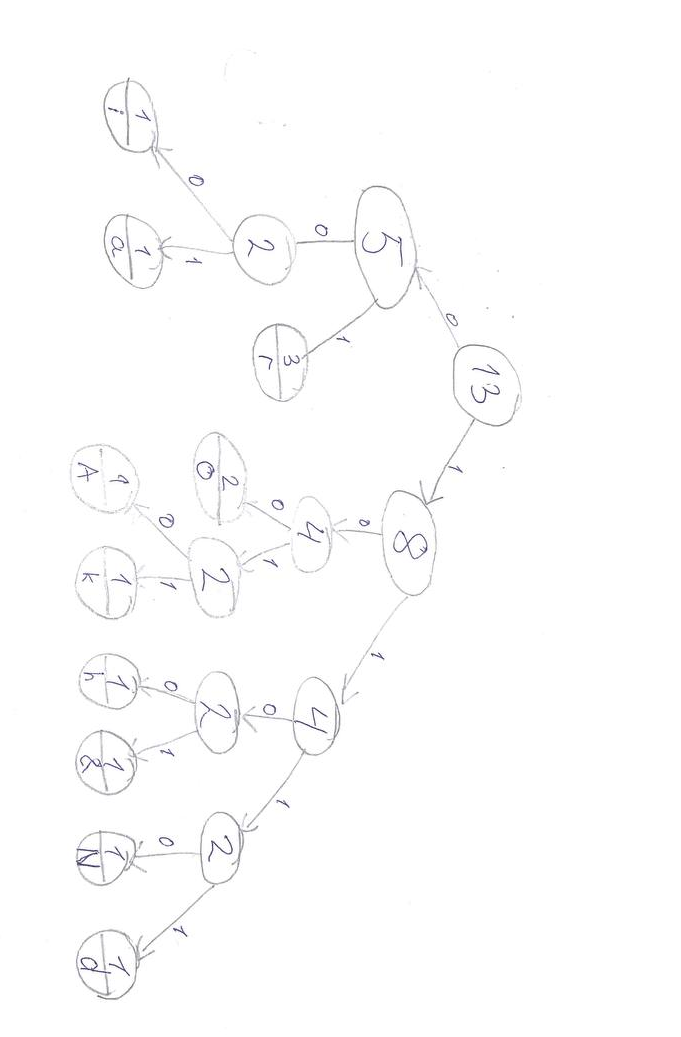
Task 3

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