1. <https://github.com/00010401/CSF.CW1.00010401>
2. a) ID number: 10401.

Converting ID number in decimal into its binary equivalent:

|  |  |  |
| --- | --- | --- |
|  | **Remainder** | **Binary** |
| 10401/2=5200 | 1 | 1 |
| 5200/2=2600 | 0 | 0 |
| 2600/2=1300 | 0 | 0 |
| 1300/2=650 | 0 | 0 |
| 650/2=325 | 0 | 0 |
| 325/2=162 | 1 | 1 |
| 162/2=81 | 0 | 0 |
| 81/2=40 | 1 | 1 |
| 40/2=20 | 0 | 0 |
| 20/2=10 | 0 | 0 |
| 10/2=5 | 0 | 0 |
| 5/2=2 | 1 | 1 |
| 2/2=1 | 0 | 0 |
| 1/2=0 | 1 | 1 |

So, the number is 10100010100001.

Converting ID number in decimal to its hex equivalent:

|  |  |  |
| --- | --- | --- |
|  | **Remainder** | **Hex** |
| 10401/16=650 | 1 | 1 |
| 650/16=40 | 10 | A |
| 40/16=2 | 8 | 8 |
| 2/16=0 | 2 | 2 |

So, the number is 28A1.

b) 1)

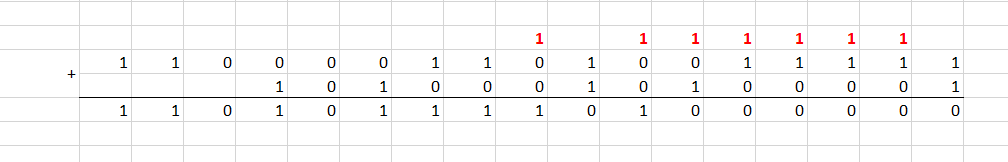
Decimal 99999 to binary:

|  |  |  |
| --- | --- | --- |
|  | **Remainder** | **Binary** |
| 99999/2=49999 | 1 | 1 |
| 49999/2=24999 | 1 | 1 |
| 24999/2=12499 | 1 | 1 |
| 12499/2=6249 | 1 | 1 |
| 6249/2=3124 | 1 | 1 |
| 3214/2=1562 | 0 | 0 |
| 1562/2=781 | 0 | 0 |
| 781/2=390 | 1 | 1 |
| 390/2=195 | 0 | 0 |
| 195/2=97 | 1 | 1 |
| 97/2=48 | 1 | 1 |
| 48/2=24 | 0 | 0 |
| 24/2=12 | 0 | 0 |
| 12/2=6 | 0 | 0 |
| 6/2=3 | 0 | 0 |
| 3/2=1 | 1 | 1 |
| 1/2=0 | 1 | 1 |

So, 99999 in decimal is equal to 11000011010011111 in binary.

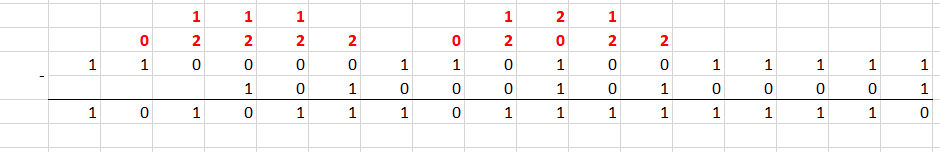
ID 10401 in decimal is equal to 10100010100001 in binary.

Decimal 99999 in binary and ID number in binary addition:



**Carry ->**

2) Decimal 99999 in binary and ID 10401 in binary subtraction:



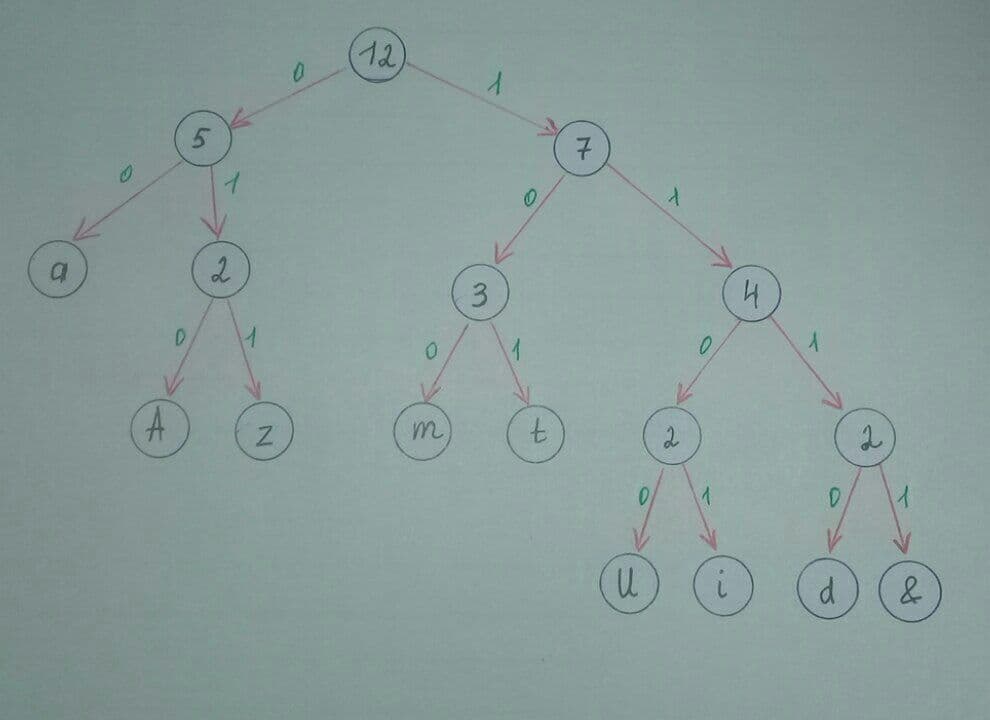
**<- Borrow**

c) Hex number system is mainly used by programmers, as it eases binary system by allowing grouping of binary digits and demonstrating long binary values in a short format, which makes it easy to understand compared to long binary values. It handles more information, however uses comparatively less storage. Moreover, hex numbers are used to represent colors in HTML and CSS; MAC addresses of devices; memory location of the errors. In hex number system it is easier to track the bugs and to debug, also it decreases the bug occurrence. (1)

1. Umida&Azamat
2. Table:

|  |  |  |
| --- | --- | --- |
| Letter | Frequency | Encoding |
| U | 1 | **1100** |
| m | 2 | **100** |
| i | 1 | **1101** |
| d | 1 | **1110** |
| a | 3 | **00** |
| & | 1 | **1111** |
| A | 1 | **010** |
| z | 1 | **011** |
| t | 1 | **101** |

1. Huffman tree:



Answer: 37bits will be needed to encode “Umida&Azamat” message using Huffman encoding.

1. a) 10401 and 45237

1, 0, 4, 0, 1, 4, 5, 2, 3, 7

Sorting: 0, 0, 1, 1, 2, 3, 4, 4, 5, 7

b) The number to look for: 5

c) Pseudo code:

1) Find midpoint:

a) Count the total number of entries in the set of numbers; (10)

b) Divide it by 2; (10/2=5)

c) 5th number will be midpoint. (2)

2) If ignore the range on the left till the midpoint, inclusively. Otherwise, ignore the range on the right side, beginning with midpoint. (As 5>2, we ignore [0-2]).

3) Find midpoint from remaining set of numbers:

a) Count the total number of entries in the set of numbers; (5)

b) Divide it by 2; (5/2=2.5)

c) Round the result 2.5≈3.

d) 3rd number will be midpoint. (4)

4) If ignore the range on the left till the midpoint, inclusively. Otherwise, ignore the range on the right side, beginning with midpoint. (As 4<5, we ignore [3-4]).

5) Find midpoint (2/2=1). 1st value is 5.

6) (x=5) = (midpoint=5), so the number is found.

1. Main memory in paged memory management consists of small fixed-size frames, and process includes pages which have an equal size with frames. In contrast, partition management technique is made up from various blocks/partitions which are mostly stored in contiguous areas of memory. (2)

In single contiguous memory management, the program should find one large piece of memory for loading data, however, in paged management technique it will search for smaller chunks of memory to load data, as it allows to divide the process into small parts.

Moreover, in all other memory management techniques the whole process of loading program should place in memory entirely, but the demand paged technique gives an opportunity to bring only needed parts of process into memory and to bring others parts when needed by swapping between main and secondary memory pages. (2)



To find physical address, according to formula we need number of frames. We can find number of frames by looking to the given table in cell respective to the page number. So, as in this case page number is 2, the number of frames will be 5 respectively.



Answer: We cannot find physical address in this case, as given offset of page overflows the frame size.

1. Software development methodologies: Waterfall and Agile comparison.

Waterfall is structured sequential attitude, while Agile consists of a lot of sprints which means a lot of iterations and testing during the development process. Meaning that Waterfall consists of only each software development stage once, but in Agile the process is divided into cycles in which the stages are repeating in each spring. Moreover, in Agile methodology the focus is more on presenting the work compared to documentation, while Waterfall cares more about structures and documentation. Agile software development methodology is structured for easy handling different changes during the development process and complex projects, whereas in Waterfall everything should be readily planned before the project starts, because it doesn’t allow changes once the process starts. In Waterfall less team collaboration/coordination is required, however, Agile needs a lot of coordination during the development and it is structured in a way that client satisfaction should be in the highest level. (3)

When budget and time is constrained, and the project has already set up requirements and well-planned scope, it is more appropriate to use Waterfall methodology, as it allows much faster delivery to meet the deadlines. Furthermore, when the project is small, requirements are not going to change it is recommended to use this methodology. (4)

In contrast, when requirements are more likely to change over the developing period, more customer involved process, flexibility for team is needed Agile methodology should be used. Moreover, it is used when the team and customers are not sure about software features, Agile is handy to use to not lose time, and in this methodology production of features is done in relatively short periods. (4)

1. In Ring topology, the computer networks are connected one to another with the help of cables and each device is connected with devices both from right and left sides, in which the data transfer is made from one device to another.

But, in Star topology, all devices are connected to each other through central hub. Meaning that the data transferring will be delivered only through central hub to other devices. (5)

|  |  |
| --- | --- |
| **Advantages of Ring Topology** | **Advantages of Star Topology** |
| * Equal access to resources between devices * Installing and expanding the network is cheap * Data transferring speed is very high * Reduced number of collisions | * Easy to install new devices, without affecting other devices performance * Fail of some devices in network doesn’t affect others and it is easy to find and troubleshoot * Easy to control the network with help of central hub |
| **Disadvantages of Ring Topology** | **Disadvantages of Star Topology** |
| * If one device connection goes down, entire network will be affected and shut down * Not easy to troubleshoot * While installing new devices the network will be interrupted | * Requires very long cables * Costly * If the central hub fails, all device in the network will go down * Number of devices that can be added to network depends on central hub capacity |

Star topology is used for connecting devices and LAN through central hub. It is widely used in larger organizations, educational fields, such as universities, colleges, where priority stands on high performance, also it is used in wi-fi. (6)

Ring topology is also used in educational institutions because of its low cost. Ring topology is also used for LAN purposes, where each device is connected with two other devices next and before it. Moreover, some organization use ring topology as a backup system. (7)

1. **def** fruits\_color(): *#this is the void function which can be only executed* print(**'blueberry is blue'**) *#and do not return the value* print(**'banana is yellow'**) *#so, from this function we* print(**'raspberry is pink'**) *#cannot return values*

*#it will return none*print(**'This is the list of fruits and their color'**)  
fruits\_color() *#in this case it only executes the function*print(**'Done!'**)  
  
  
animals = [**'lion'**, **'penguin'**, **'rabbit'**] *#this is a list*type = (**'wild'**, **'mammal'**, **'domestic'**) *#this is a tuple***def** animals\_type(animals, type): *#this function returns a* animals\_type = tuple(zip(animals, type)) *#value, as it has return*

*#statement*  
 **return** animals\_type *#this is the return statement*print(animals\_type(animals, type))

*#this is an example of object oriented programming  
#encapsulates functionality using classes***class** Fruit: *#Self is automatically assigned to the new piece of Fruit class* **def** \_\_init\_\_(self, name, colour):  
 self.name = name  
 self.colour = colour  
  
 **def** fruits\_color(self):  
 fruits\_color = self.name, self.colour  
 **return** fruits\_color  
  
 **def** displayFruits(self):  
 print(self.fruits\_color())  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 fruit1 = Fruit(**'kiwi'**, **'green'**)  
 fruit1.displayFruits()  
  
 fruit2 = Fruit(**'pomegranate'**, **'red-pink'**)  
 fruit2.displayFruits()  
  
  
*#functional approach  
#pure functions will be used***def** recipe\_function(Recipe): *#here the Recipe function will not #change the input recipe* new\_recipe = [] *#however will return new recipe* **for** i **in** Recipe:  
 new\_recipe.append(i+**'+sugar'**)  
 **return** new\_recipe  
  
raw\_Recipe = [**'coffee'**, **'tea'**, **'lemonade'**]  
sweet\_Recipe = recipe\_function(raw\_Recipe)  
  
print(raw\_Recipe)  
print(sweet\_Recipe)

**Reference list:**

1. Uses of Hexadecimal [Internet]. Computer Science. 2019 [cited 2021 Jan 14]. Available from: <https://www.computerscience.gcse.guru/theory/uses-of-hexadecimal>
2. Dale N, Lewis J. Computer science illuminated. 7th ed. Burlington, Massachusetts: Jones & Bartlett Learning; 2020.
3. ALTVATER A. What is Agile Methodology? How It Works, Best Practices, Tools [Internet]. stackfy.com. 2017 [cited 2021 Jan 15]. Available from: <https://stackify.com/agile-methodology/#:~:text=Examples%20of%20Agile%20Methodology,Lean%20Software%20Development%20(LSD).&text=They%20assess%20progress%20in%20a%20meeting%20called%20a%20daily%20scrum>.
4. 4.Parsons T. When to Use Waterfall vs. Agile [Internet]. Macadamian. 2019 [cited 2021 Jan 15]. Available from: <https://www.macadamian.com/learn/when-to-use-waterfall-vs-agile/#:~:text=When%20to%20Use%20the%20Waterfall>
5. Circuit Globe. Difference Between Star and Ring Topology [Internet]. Circuit Globe. 2019 [cited 2021 Jan 15]. Available from: https://circuitglobe.com/difference-between-star-and-ring-topology.html
6. Star networks - Network topologies, protocols and layers - OCR - GCSE Computer Science Revision - OCR [Internet]. BBC Bitesize. 2019 [cited 2021 Jan 15]. Available from: <https://www.bbc.co.uk/bitesize/guides/zr3yb82/revision/1#:~:text=Star%20topologies%20are%20used%20in>
7. What is Ring Topology? | How it Works | Use & Benefits of Ring Topology [Internet]. EDUCBA. 2020 [cited 2021 Jan 15]. Available from: https://www.educba.com/what-is-ring-topology/

‌

‌