A1 Team 21

**Algorithms in English**

**#Problem 1**

1. To find maximum occurrence element :

* Assign four variables to count and keep track of max element and it’s occurrence.
* Iterate through the list
* Iterate through the list inside the other loop to calculate the occurrence of each element
* Compare with occurrence of most occurring element

1. To find minimum occurrence element :

* Assign four variables to count and keep track of max element and it’s occurrence.
* Iterate through the list
* Iterate through the list inside the other loop to calculate the occurrence of each element
* Compare with occurrence of least occurring element

1. Return least occurring element

**#Problem 2**

1. Send the function the list and size of window
2. Create a new empty list to make sure the original list does not get changed
3. Set a temporary list to size of window
4. Find maximum element of window temporary list
5. Max element of window list becomes the first element of the list
6. List is stopped at Lenth of list – window size

**PSEUDOCODE**

**#Full Name: Kartikey Pandey**

**#Team Members: Kartikey Pandey(kfp5411), Riya Rawal(rzr5540), Aviral Bansal(akb6824)**

**#ID:939970318**

**#Date:28th September 2022**

**#Filename:A1\_Team21\_Pandey\_Kartikey\_kfp5411**

**#Purpose: Code FOR A1 FOR Team 21**

**#Problem 1**

**#Create a function that, given a list of unknown size, RETURNs its most repeated element and its least repeated element as a list of the two elements IN that order. Assign the RETURNED list to a variable and then OUTPUT the variable on another line. You can test the function by giving it [ 10, 30, 60, 88, 10, 30, 10, 60, 3, 88 ], the RETURNED list FOR which should be [ 10, 3 ].**

**DEFINE FUNCTION ml(l):**

**#Counting Max**

**SET counter TO 0**

**SET count TO 0**

**SET word TO ""**

**SET word\_max TO “”**

**FOR i IN range(len(l)):**

**SET word TO l[i]**

**FOR j IN range(len(l)):**

**IF l[j] EQUALS word:**

**count += 1**

**IF count >= counter:**

**SET counter TO count**

**word\_max= l[i]**

**SET count TO 0**

**SET counter\_2 TO 1**

**SET count\_2 TO 0**

**SET word\_2 TO ""**

**SET word\_min TO ""**

**FOR i IN range(len(l)):**

**SET word\_2 TO l[i]**

**FOR j IN range(len(l)):**

**IF l[j] EQUALS word\_2:**

**count\_2 += 1**

**IF count\_2 <= counter\_2:**

**SET counter\_2 TO count\_2**

**word\_min= l[i]**

**SET count\_2 TO 0**

**RETURN word\_max,word\_min**

**#Problem 2**

**#Consider a list and another variable called window size(w). Let us define a sliding window as a part of the list of size w. This window keeps sliding one element at a time from the very left. Your task is to create a resulting array that contains the maximum of each of the sliding windows.**

**DEFINE FUNCTION max(l):**

**word\_max= l[0]**

**FOR i IN range(len(l)):**

**IF l[i] >= word\_max:**

**SET word\_max TO l[i]**

**RETURN word\_max**

**DEFINE FUNCTION new\_list(ls,i,w):**

**SET nl TO []**

**FOR i IN range(i,w+i):**

**SET nl TO nl + [ls[i]]**

**SET i TO i + 1**

**RETURN nl**

**DEFINE FUNCTION sliding\_window(ls, w):**

**FOR i IN range(0,len(ls)-w+1):**

**SET k TO new\_list(ls,i,w)**

**SET g TO max(k)**

**SET ls[i] TO g**

**SET final TO []**

**FOR d IN range(0,len(ls)-w+1):**

**SET final TO final + [ls[d]]**

**RETURN final**

**#Main function**

**DEFINE FUNCTION main():**

**#Problem 1**

**SET l TO [1,2,3,4,5,6,7,8,9,1,2,3,4,5,6,7,8,1,2,3,4,5,6,7,1,2,3,4,5,6,1,2,3,4,5,1,2,3,4,1,2,3,1,2,1]**

**SET prob\_1 TO ml(l)**

**OUTPUT("Most occuring element =",prob\_1[0])**

**OUTPUT("Least occuring element =",prob\_1[1])**

**#Problem 2**

**SET l TO [1,3,5,1,2,4,7,8]**

**SET w TO 4**

**SET prob\_2 TO sliding\_window(l,w)**

**OUTPUT("Output FOR Problem 2 =",prob\_2)**

**#Call FOR main function**