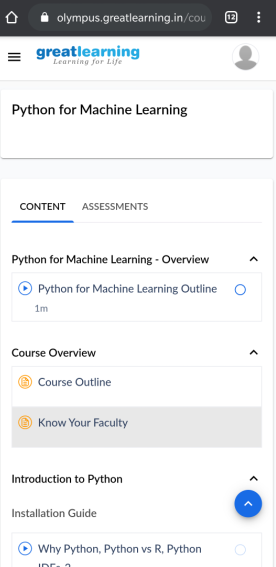
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **20-06-2020** | | | | | **Name:** | **Rakesh M Kotian** | |
| **Sem & Sec** | **8 th sec-b** | | | | | **USN:** | **4al16cs072** | |
| **Online Test Summary** | | | | | | | | |
| **Subject** | | **-** | | | | | | |
| **Max. Marks** | | **-** | | **Score** | | | **-** | |
| **Certification Course Summary** | | | | | | | | |
| **Course** | **Python for machine learning** | | | | | | | |
| **Certificate Provider** | | | **Great learning** | | **Duration** | | | **6 hours** |
| **Coding Challenges** | | | | | | | | |
| **Problem Statement:**  Jump search | | | | | | | | |
| **Status:solved** | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | | |
| **If yes Repository name** | | | | | **Rakeshkotian08** | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | |

Online Test Details: (Attach the snapshot and briefly write the report for the same)

Certification Course Details: (Attach the snapshot and briefly write the report for the same)



|  |  |
| --- | --- |
| |  | | --- | | // C++ program to implement Jump Search    #include <bits/stdc++.h>  using namespace std;    int jumpSearch(int arr[], int x, int n)  {      // Finding block size to be jumped      int step = sqrt(n);        // Finding the block where element is      // present (if it is present)      int prev = 0;      while (arr[min(step, n)-1] < x)      {          prev = step;          step += sqrt(n);          if (prev >= n)              return -1;      }        // Doing a linear search for x in block      // beginning with prev.      while (arr[prev] < x)      {          prev++;            // If we reached next block or end of          // array, element is not present.          if (prev == min(step, n))              return -1;      }      // If element is found      if (arr[prev] == x)          return prev;        return -1;  }    // Driver program to test function  int main()  {      int arr[] = { 0, 1, 1, 2, 3, 5, 8, 13, 21,                  34, 55, 89, 144, 233, 377, 610 };      int x = 55;      int n = sizeof(arr) / sizeof(arr[0]);        // Find the index of 'x' using Jump Search      int index = jumpSearch(arr, x, n);        // Print the index where 'x' is located      cout << "\nNumber " << x << " is at index " << index;      return 0;  } |   {      int arr[] = {1, 2, 3, 4, 5}, i;      int n = sizeof(arr) /              sizeof(arr[0]);        cout << "Given array is \n";      for (i = 0; i < n; i++)          cout << arr[i];        rotate(arr, n);        cout << "\nRotated array is\n";      for (i = 0; i < n; i++)          cout << arr[i];        return 0;  } |