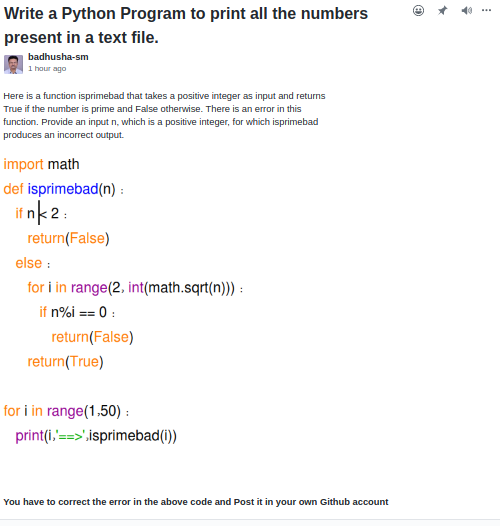
**DAILY ONLINE ACTIVITIES SUMMARY**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | | **19.5.20** | | | | | |
| **Name of the Faculty** | | **Dr.S.Mohideen Badhusha** | | | | | |
| **No. of Students Allotted for monitoring** | | | | | **18** | | |
| **Progress** | | | | | **The google doc file is sent to each one of them to collect their github account details** | | |
| **Whether collected report from all the allotted students** | | | | | **11 members have responded and sent their github account details. 7 members have been intimated to send the account details immediately** | | |
| **Online Test Summary** | | | | | | | |
| **Subject** |  | | | | | | |
| **Semester** |  | | | **Duration** | |  | |
| **No. of students Taken** | | |  | **Passing %** | |  | |
| **Certification Course Summary** | | | | | | | |
| **Course** | | **Not yet updated by the students. Asked them to update immediately** | | | | | |
| **Certificate Provider** | |  | | | **Duration** | |  |
| **Coding Challenges** | | | | | | | |
| **Problem Statement: Two python assignments have been posted in github** | | | | | | | |
| **Status: so far there is no submission** | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes** | | |
| **If yes Repository name** | | | | | **CSE** | | |
| **Uploaded the report in slack** | | | | | **CSE** | | |

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

Coding Challenges Details: (Attach the snapshot and briefly write the report for the same)



Out of 18 , the following 13 students have given github account information

VI Semester students to be monitored

Mentor Name : Dr.Sm.Badhusha, Prof/CSE, Alva’s Institute of Engineering and Technology

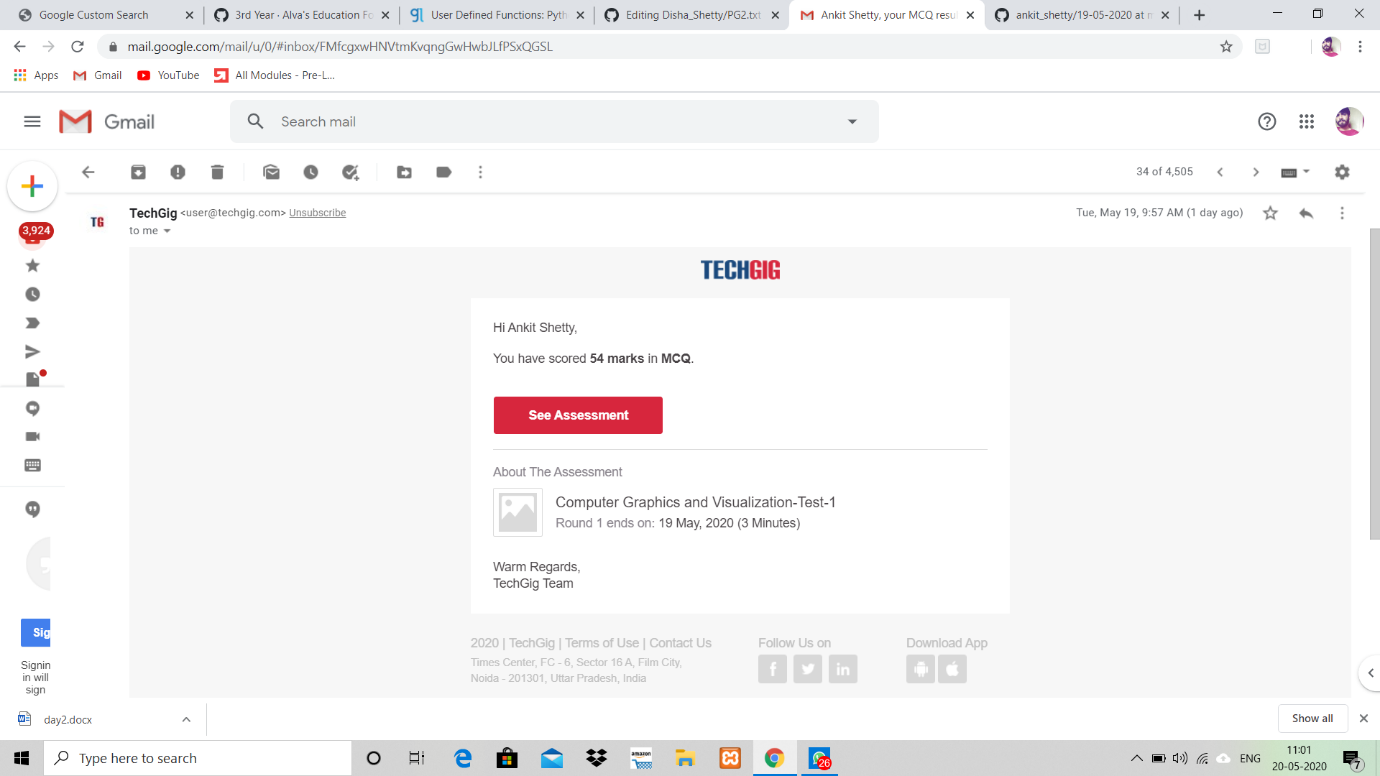
|  |  |  |
| --- | --- | --- |
|  | 4AL17CS086 | Shetty Ankit Suresh  https://github.com/alvas-education-foundation/ankit\_shetty |
|  | 4AL17CS087 | Shetty Disha Ravindra  <https://github.com/alvas-education-foundation/Disha_Shetty> |
|  | 4AL17CS089 | Shetty Sathvik Ravindra  <https://github.com/sathvikshetty22> |
|  | 4AL17CS090 | Shilpa S U  <http://gihub.com/alvas-education-foundation/Shilpa-S.U> |
|  | 4AL17CS091 | Shreetal Kalabandi  https://github.com/Shreetal |
|  | 4AL17CS092 | Shrinivasa |
|  | 4AL17CS093 | Shwetha M S  https://github.com/alvas-education-foundation/Shwetha |
|  | 4AL17CS094 | Sindhu N  https://github.com/alvas-education-foundation/Sindhu-N |
|  | 4AL17CS095 | Sneha Shetty Disha RavindraK Bakale  https://github.com/alvas-education-foundation/Sneha-K-Bakale.git |
|  | 4AL17CS096 | Soundarya R |
|  | 4AL17CS097 | Spoorthi M S |
|  | 4AL17CS098 | Spoorthi Balaji  https://github.com/spoorthybalaji |
|  | 4AL17CS099 | Shrilatha K Kamath  https://github.com/alvas-education-foundation/Srilatha-K-Kamat |
|  | 4AL17CS100 | Suhas M S |
|  | 4AL17CS101 | Surya Prakash S |
|  | 4AL17CS102 | Sushmitha  <https://github.com/sushmithaganiga/progress> |
|  | 4AL17CS103 | Sushmitha B Poojary  https://github.com/alvas-education-foundation/sushmitha-b-poojary |
|  | 4AL17CS104 | Syed Hudaif Ibrahim  [https://github.com/SyedHudaif](https://github.com/alvas-education-foundation/Syed-Hudaif) |

shetty Ankit

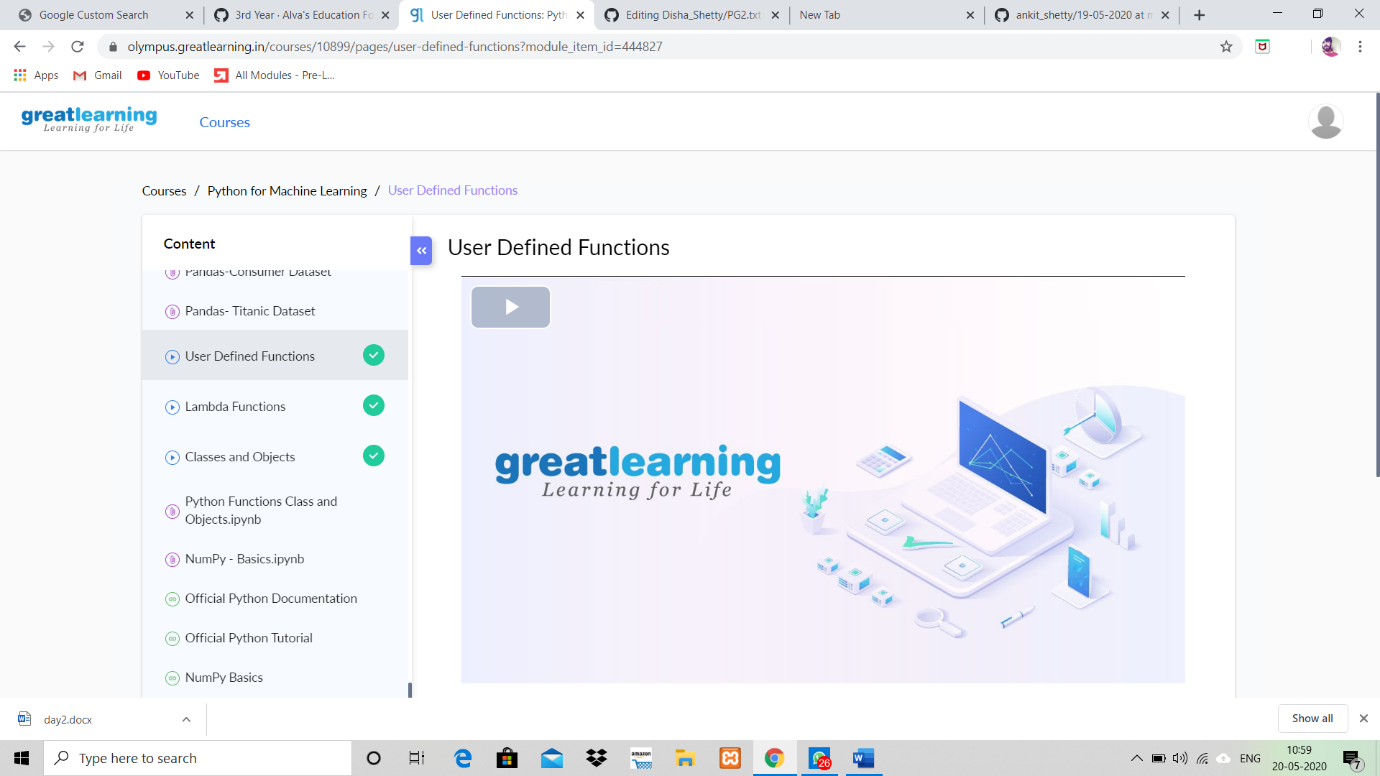
online coding

|  |  |
| --- | --- |
| program 1 |  |
|  | import java.util.Stack; |
|  | public class Main { |
|  | public static void main(String[] a){ |
|  | Node n1 = new Node(10); |
|  | Node n2 = new Node(28); |
|  | Node n3 = new Node(15); |
|  | Node n4 = new Node(29); |
|  | Node n5 = new Node(10); |
|  | n1.next = n2; |
|  | n2.next = n3; |
|  | n3.next = n4; |
|  | n4.next = n5; |
|  | boolean result = isPalindrome(n1); |
|  | System.out.println("Is it palindrome: "+result); |
|  | } |
|  | static class Node { |
|  | int data; |
|  | Node next; |
|  | Node(int tmp) {Shetty Disha Ravindra |
|  | data = tmp;Shetty Disha Ravindra |
|  | } |
|  | } |
|  |  |
|  | program 2 |
|  |  |
|  | #include <stdio.h> |
|  | #include <string.h> |
|  | int check\_subsequence (char [], char[]); |
|  | int main () { |
|  | int flag; |
|  | char s1[1000], s2[1000]; |
|  | printf("Input first string\n"); |
|  | gets(s1); |
|  | printf("Input second string\n"); |
|  | gets(s2); |
|  | if (strlen(s1) < strlen(s2)) |
|  | flag = check\_subsequence(s1, s2); |
|  | else |
|  | flag = check\_subsequence(s2, s1); |
|  | if (flag) |
|  | printf("YES\n"); |
|  | else |
|  | printf("NO\n"); |
|  | return 0; |
|  | } |
|  | int check\_subsequence (char a[], char b[]) { |
|  | int c, d; |
|  | c = d = 0; |
|  | while (a[c] != '\0') { |
|  | while ((a[c] != b[d]) && b[d] != '\0') { |
|  | d++; |
|  | } |
|  | if (b[d] == '\0') |
|  | break; |
|  | d++; |
|  | c++; |
|  | } |
|  | if (a[c] == '\0') |
|  | return 1; |
|  | else |
|  | return 0; |
|  | } |

online test



online course



**Shetty Disha Ravindra**

**online coding**

|  |
| --- |
| import java.util.Stack; |
| public class Main { |
| public static void main(String[] a){ |
| Node n1 = new Node(10); |
| Node n2 = new Node(28); |
| Node n3 = new Node(15); |
| Node n4 = new Node(29); |
| Node n5 = new Node(10); |
| n1.next = n2; |
| n2.next = n3; |
| n3.next = n4; |
| n4.next = n5; |
| boolean result = isPalindrome(n1); |
| System.out.println("Is it palindrome: "+result); |
| } |
| static class Node { |
| int data; |
| Node next; |
| Node(int tmp) { |
| data = tmp; |
| } |

}

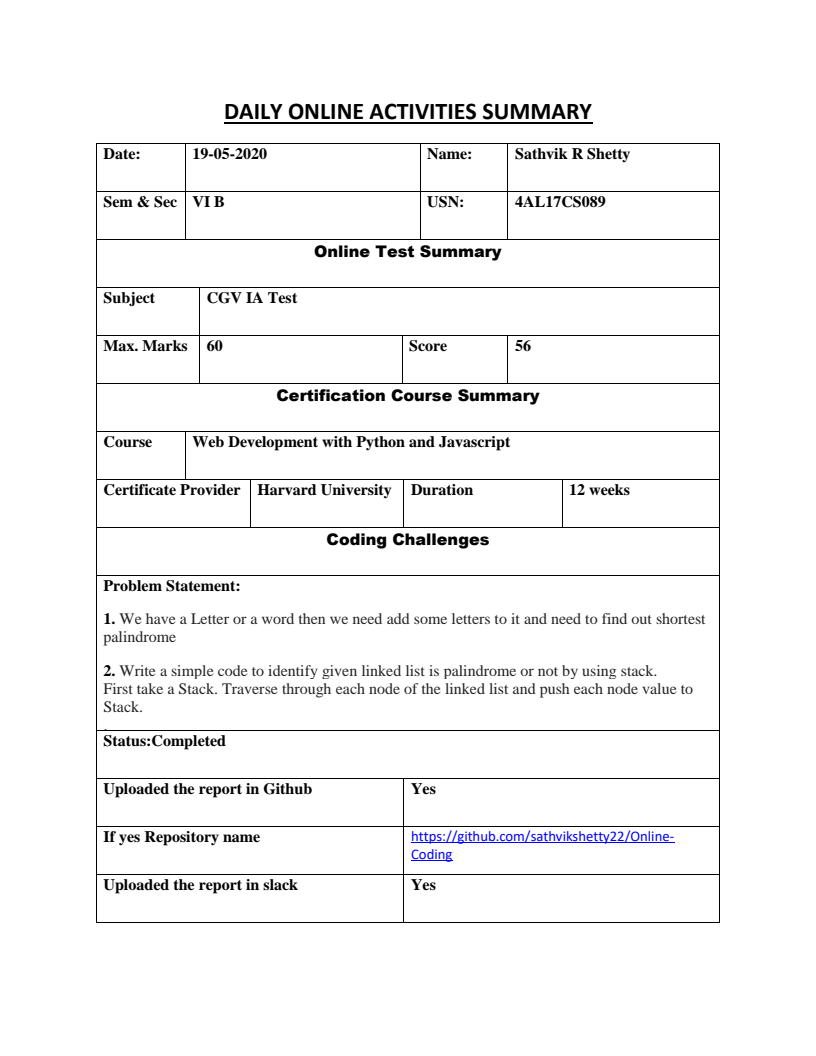
online test

nil

online course

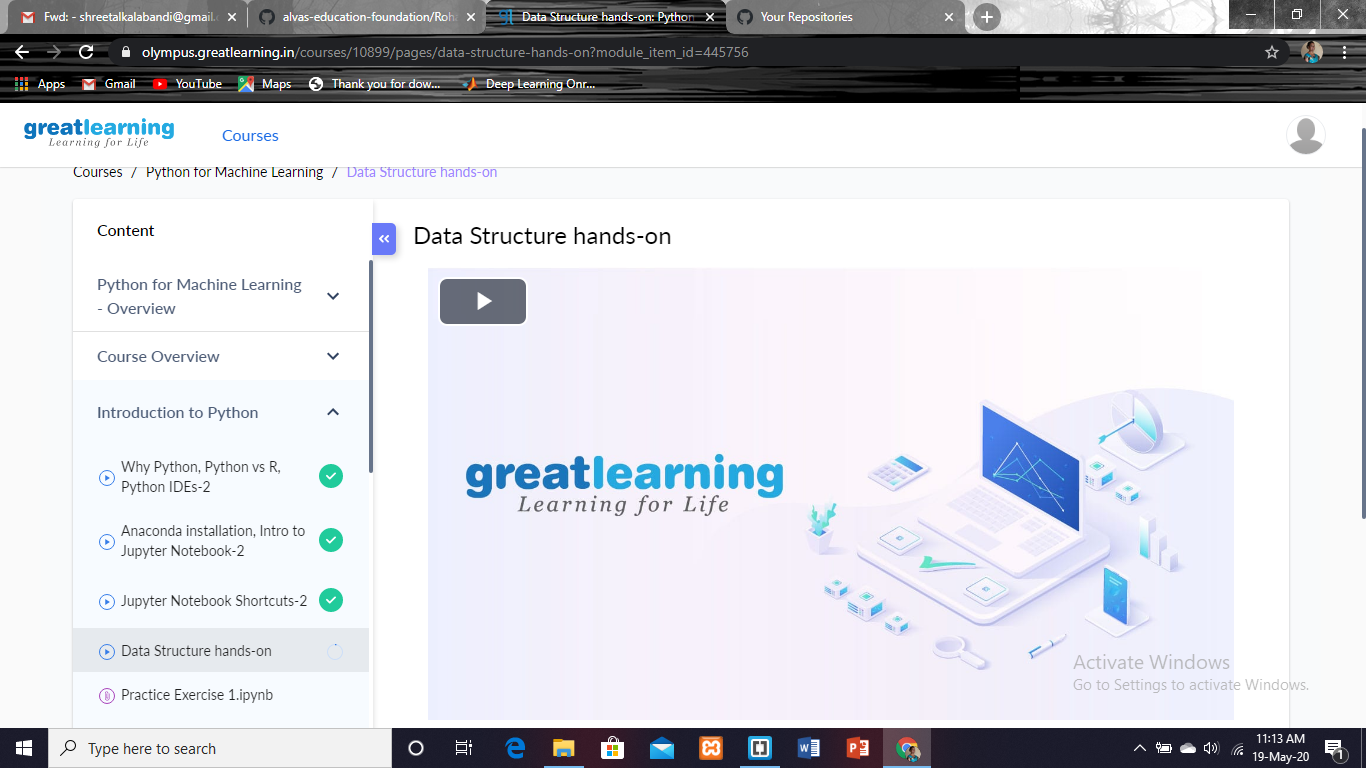
nil

Shetty Sathvik Ravindra



Shreetal Kalabandi

online course



online activities

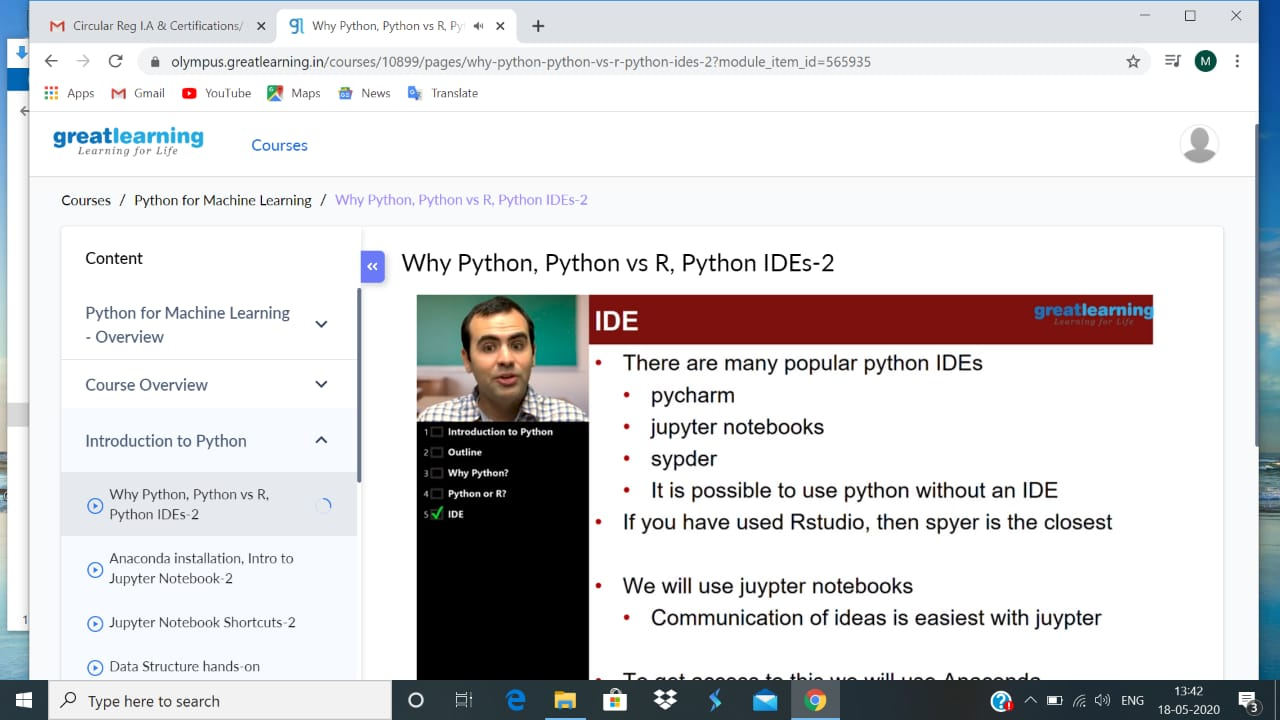
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Date:** | **18th may 2020** | | | | | | **Name:** | | **Shreetal Kalabandi** | |
| **Sem & Sec** | **6th sem ‘B’** | | | | | | **USN:** | | **4al17cs091** | |
| **Online Test Summary** | | | | | | | | | | |
| **Subject** | | **CNSC** | | | | | | | | |
| **Max. Marks** | | **60** | | **Score** | | | | **43** | | |
| **Certification Course Summary** | | | | | | | | | | |
| **Course** | **Great Learning(Python for machine learning)** | | | | | | | | | |
| **Certificate Provider** | | | **Great Learning** | | | **Duration** | | | | **5 hrs** |
| **Coding Challenges** | | | | | | | | | | |
| **Problem Statement: factorial function.** | | | | | | | | | | |
| **Status:completed.** | | | | | | | | | | |
| **Uploaded the report in Github** | | | | | **yes uploaded** | | | | | |
| **If yes Repository name** | | | | | **18thmayshreetal-4al17cs091** | | | | | |
| **Uploaded the report in slack** | | | | | **yes** | | | | | |

**Swetha MS**

**coding**

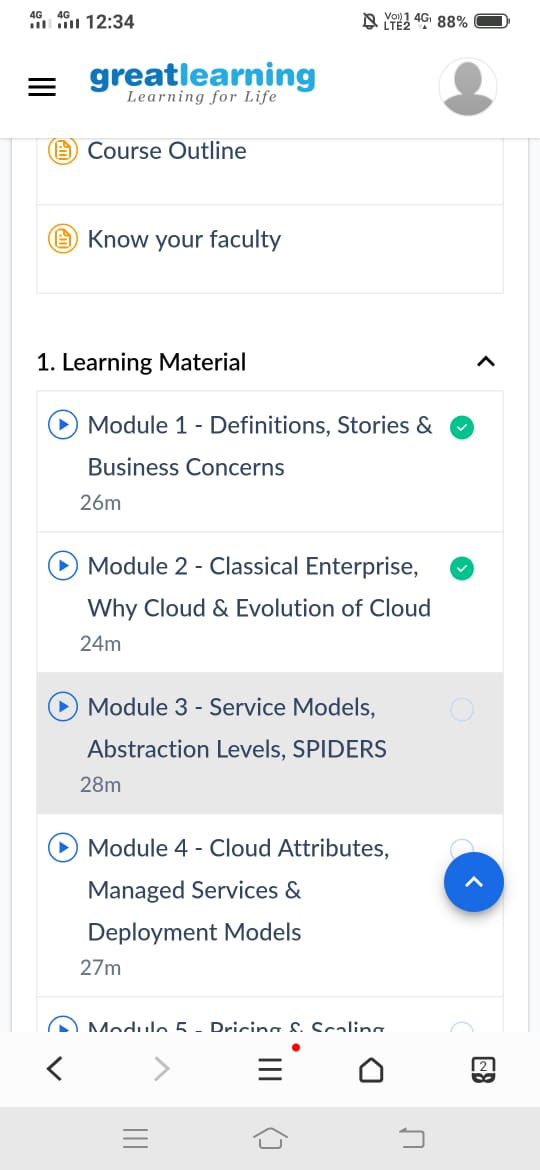
|  |
| --- |
| Program1 |
|  |
| import java.util.Scanner; |
| public class Main |
| { |
| public static void main(String args[]) |
| { |
| int i; |
| String str; |
| int counter[] = new int[256]; |
| Scanner in = new Scanner(System.in); |
| System.out.print("Enter a String : "); |
| str=in.nextLine(); |
| for (i = 0; i < str.length(); i++) { |
| counter[(int) str.charAt(i)]++; |
| } |
| // Print Frequency of characters |
| for (i = 0; i < 256; i++) { |
| if (counter[i] != 0) { |
| System.out.println( (char) i + ":" +counter[i]); |
| } |
| } |
| } |
| } |
|  |
|  |
| Program2 |
|  |
| class OddThread extends Thread |
| { |
| int limit; |
| sharedPrinter printer; |
| public OddThread(int limit, sharedPrinter printer) |
| { |
| this.limit = limit; |
| this.printer = printer; |
| } |
| @Override |
| public void run() |
| { |
| int oddNumber = 1; |
| while (oddNumber <= limit) |
| { |
| printer.printOdd(oddNumber); |
| oddNumber = oddNumber + 2; |
| } |
| } |
| } |
|  |
| class EvenThread extends Thread |
| { |
| int limit; |
| sharedPrinter printer; |
| public EvenThread(int limit, sharedPrinter printer) |
| { |
| this.limit = limit; |
| this.printer = printer; |
| } |
| @Override |
| public void run() |
| { |
| int evenNumber = 2; |
| while (evenNumber <= limit) |
| { |
| printer.printEven(evenNumber); |
| evenNumber = evenNumber + 2; |
| } |
| } |
| } |
| class sharedPrinter |
| { |
|  |
| boolean isOddPrinted = false; |
|  |
|  |
| synchronized void printOdd(int number) |
| { |
| while (isOddPrinted) |
| { |
| try |
| { |
| wait(); |
| } |
| catch (InterruptedException e) |
| { |
| e.printStackTrace(); |
| } |
| } |
| System.out.println(Thread.currentThread().getName()+" "+number); |
| isOddPrinted = true; |
| try |
| { |
| Thread.sleep(1000); |
| } |
| catch (InterruptedException e) |
| { |
| e.printStackTrace(); |
| } |
| notify(); |
| } |
|  |
| synchronized void printEven(int number) |
| { |
| while (! isOddPrinted) |
| { |
| try |
| { |
| wait(); |
| } |
| catch (InterruptedException e) |
| { |
| e.printStackTrace(); |
| } |
| } |
| System.out.println(Thread.currentThread().getName()+" "+number); |
| isOddPrinted = false; |
| try |
| { |
| Thread.sleep(1000); |
| } |
| catch (InterruptedException e) |
| { |
| e.printStackTrace(); |
| } |
| notify(); |
| } |
| } |
| public class Main |
| { |
| public static void main(String[] args) |
| { |
| sharedPrinter printer = new sharedPrinter(); |
| OddThread oddThread = new OddThread(20, printer); |
| oddThread.setName("—-pong"); |
| EvenThread evenThread = new EvenThread(20, printer); |
| evenThread.setName("ping — >"); |
| oddThread.start(); |
| evenThread.start(); |
| } |
| } |
|  |

Online course



Sindu N

Online course

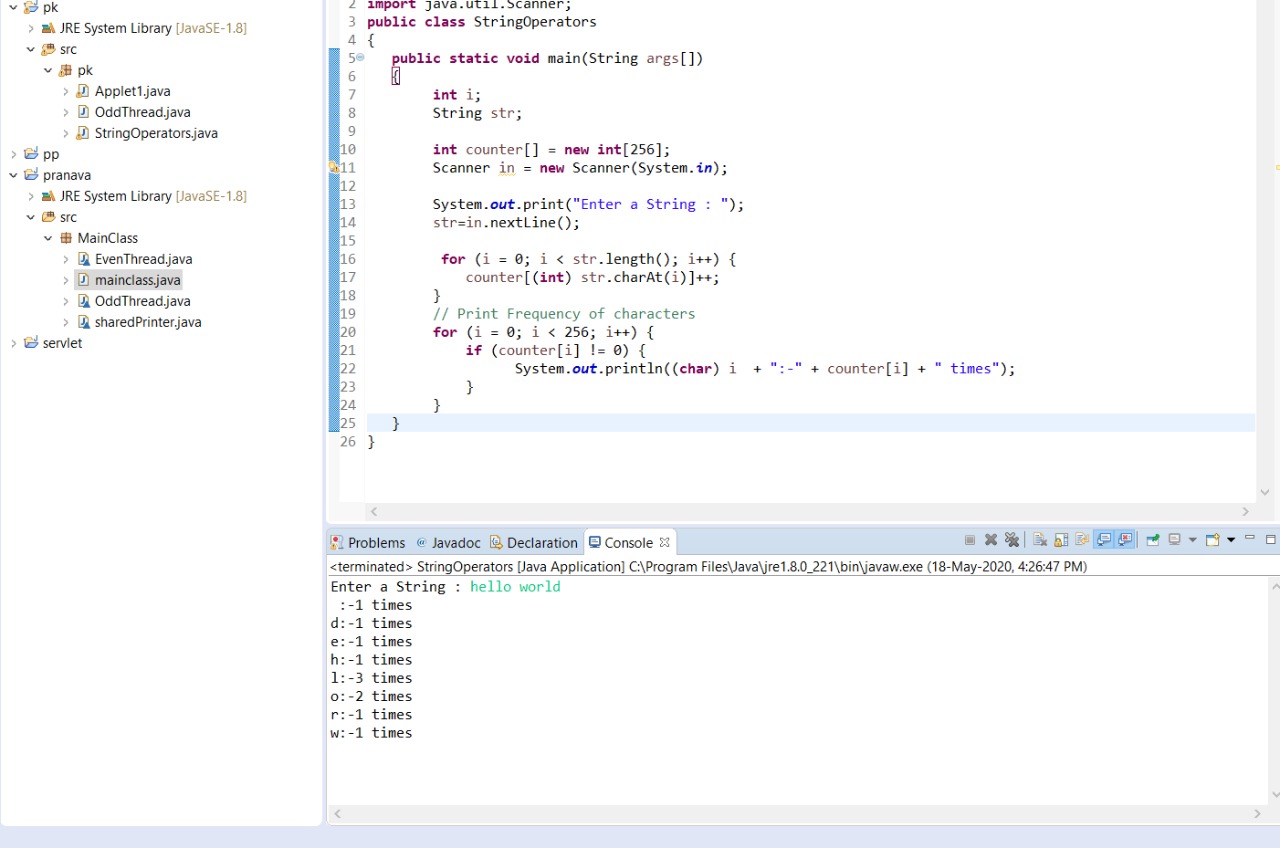


coding

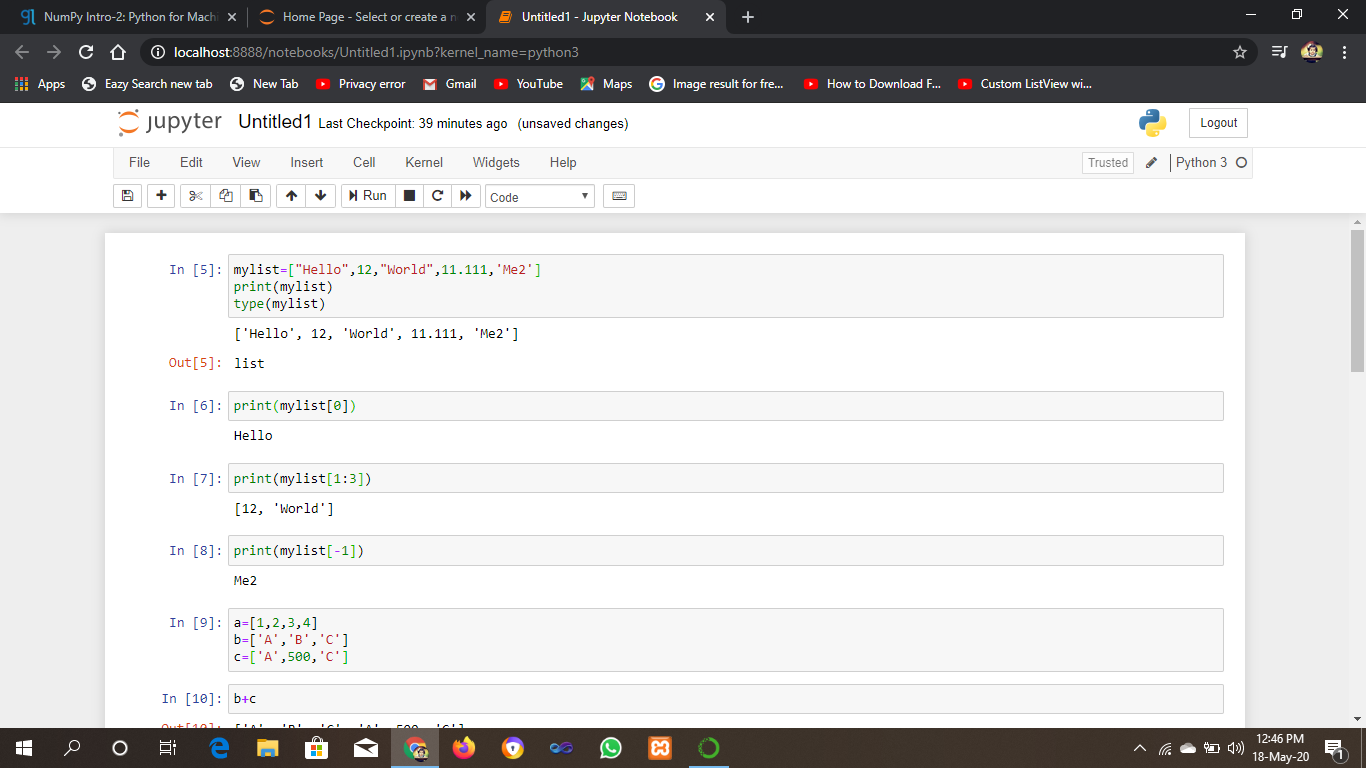
|  |
| --- |
| program1: |
|  |
|  |
| import java.util.Stack; |
|  |
| // Data Structure to store a linked list node |
| class Node { |
| int data; |
| Node next; |
|  |
| Node(int i) |
| { |
| this.data = i; |
| this.next = null; |
| } |
| }; |
|  |
| class Main |
| { |
| // Function to determine if a given linked list is palindrome or not |
| public static boolean isPalindrome(Node head) |
| { |
| // construct an empty stack |
| Stack<Integer> s = new Stack<>(); |
|  |
| // push all elements of the linked list into the stack |
| Node node = head; |
| while (node != null) { |
| s.push(node.data); |
| node = node.next; |
| } |
|  |
| // traverse the linked list again |
| node = head; |
| while (node != null) |
| { |
| // pop the top element from the stack |
| int top = s.pop(); |
|  |
| // compare the popped element with current node's data |
| // return false if mismatch happens |
| if (top != node.data) { |
| return false; |
| } |
|  |
| // advance to the next node |
| node = node.next; |
| } |
|  |
| // we reach here only when the linked list is palindrome |
| return true; |
| } |
|  |
| public static void main(String[] args) |
| { |
| Node head = new Node(1); |
| head.next = new Node(2); |
| head.next.next = new Node(3); |
| head.next.next.next = new Node(2); |
| head.next.next.next.next = new Node(1); |
|  |
| if (isPalindrome(head)) { |
| System.out.print("Linked List is a palindrome."); |
| } else { |
| System.out.print("Linked List is not a palindrome."); |
| } |
| } |
| } |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| program2: |
|  |
|  |
| package shortestpalindromeexample.java; |
| import java.util.Scanner; |
|  |
| public class ShortestPalindromeDemo { |
|  |
| public static String shortestPalindrome(String str) { |
|  |
| int x=0; |
| int y=str.length()-1; |
|  |
| while(y>=0){ |
| if(str.charAt(x)==str.charAt(y)){ |
| x++; |
| } |
| y--; |
| } |
|  |
| if(x==str.length()) |
| return str; |
|  |
| String suffix = str.substring(x); |
| String prefix = new StringBuilder(suffix).reverse().toString(); |
| String mid = shortestPalindrome(str.substring(0, x)); |
|  |
| return prefix+mid+suffix; |
| } |
|  |
| public static void main(String[] args) { |
|  |
| Scanner in = new Scanner(System.in); |
|  |
| System.out.println("Enter a String to find out shortest palindrome"); |
|  |
| String str=in.nextLine(); |
|  |
| System.out.println("Shortest palindrome of "+str+" is "+shortestPalindrome(str)); |
|  |
| } |

**Spoorthi**

**coding**

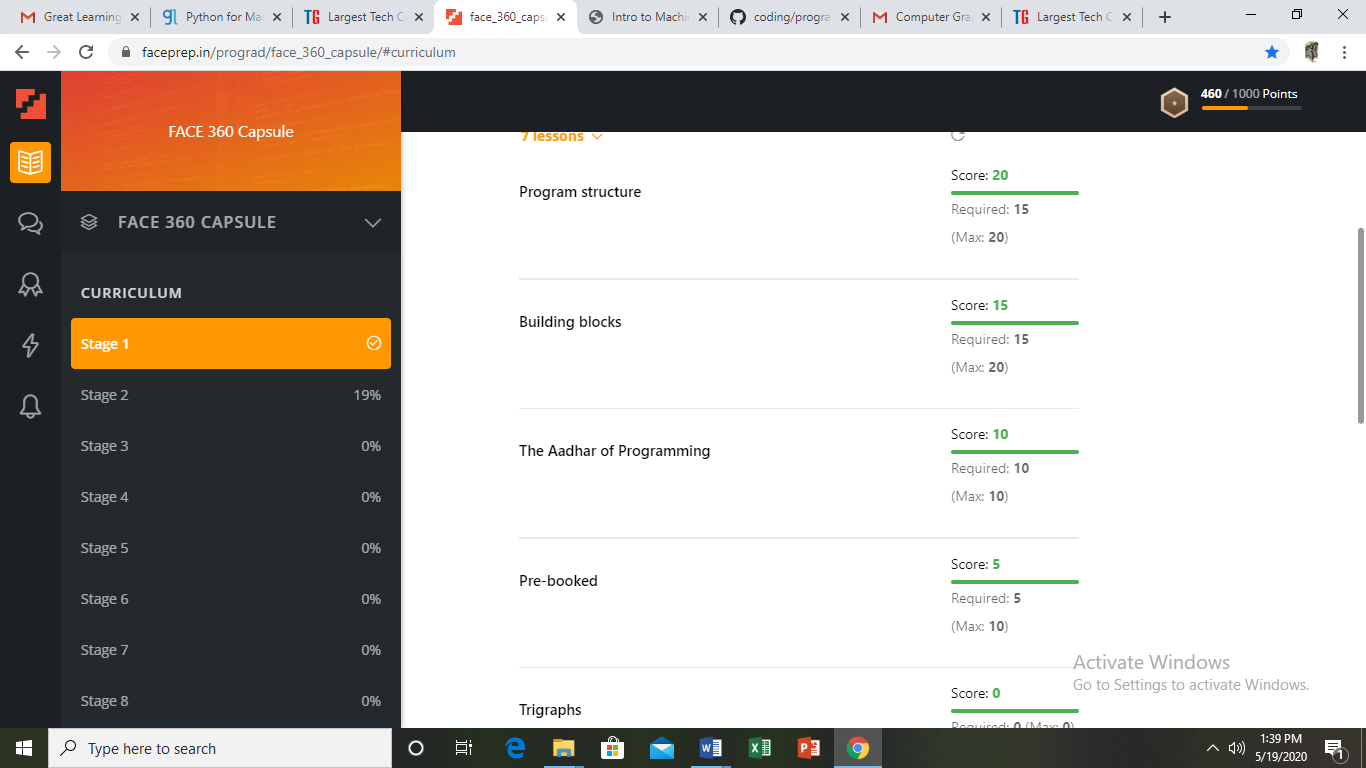


online course



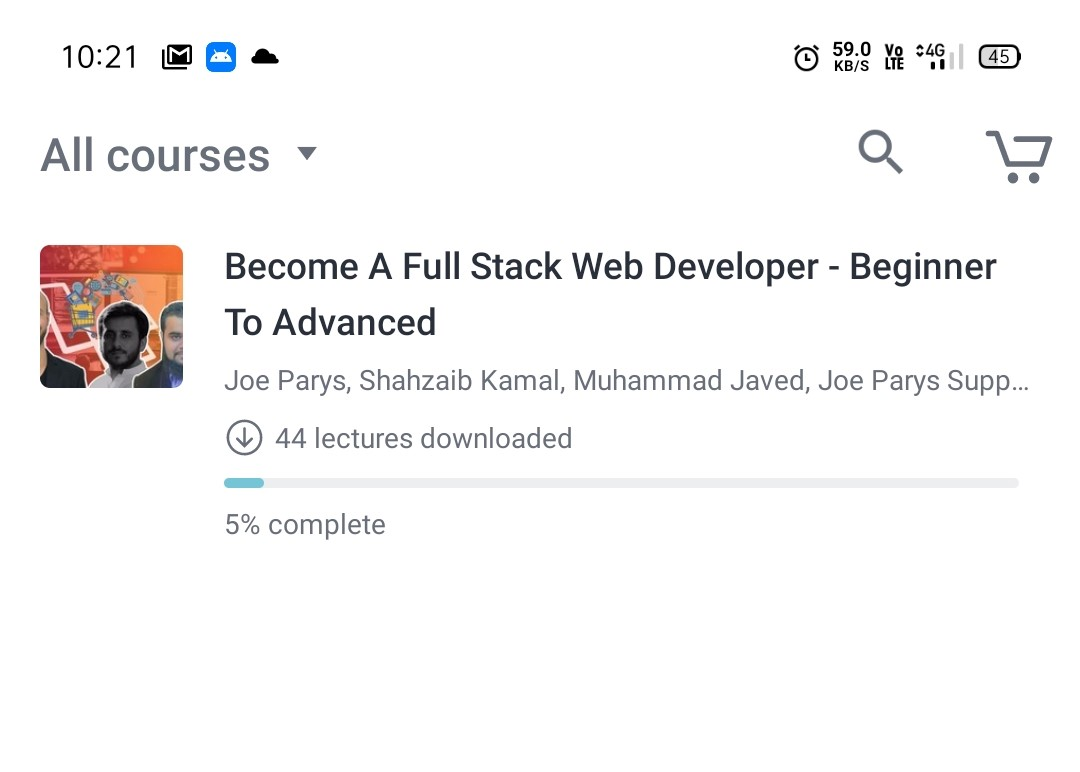
sushmitha ganiga

online course



**Syed Hudaif**

**onlne course**



coding

|  |
| --- |
| #Program 1 |
|  |
| import java.util.Scanner; |
| public class Main |
| { |
| public static void main(String args[]) |
| { |
| int i; |
| String str; |
| int counter[] = new int[256]; |
| Scanner in = new Scanner(System.in); |
| System.out.print("Enter a String : "); |
| str=in.nextLine(); |
| for (i = 0; i < str.length(); i++) { |
| counter[(int) str.charAt(i)]++; |
| } |
| // Print Frequency of characters |
| for (i = 0; i < 256; i++) { |
| if (counter[i] != 0) { |
| System.out.println( (char) i + ":" +counter[i]); |
| } |
| } |
| } |
| } |
|  |
|  |
|  |
|  |
| #Program 2 |
|  |
| class OddThread extends Thread |
| { |
| int limit; |
| sharedPrinter printer; |
| public OddThread(int limit, sharedPrinter printer) |
| { |
| this.limit = limit; |
| this.printer = printer; |
| } |
| @Override |
| public void run() |
| { |
| int oddNumber = 1; |
| while (oddNumber <= limit) |
| { |
| printer.printOdd(oddNumber); |
| oddNumber = oddNumber + 2; |
| } |
| } |
| } |
|  |
| class EvenThread extends Thread |
| { |
| int limit; |
| sharedPrinter printer; |
| public EvenThread(int limit, sharedPrinter printer) |
| { |
| this.limit = limit; |
| this.printer = printer; |
| } |
| @Override |
| public void run() |
| { |
| int evenNumber = 2; |
| while (evenNumber <= limit) |
| { |
| printer.printEven(evenNumber); |
| evenNumber = evenNumber + 2; |
| } |
| } |
| } |
| class sharedPrinter |
| { |
| boolean isOddPrinted = false; |
| synchronized void printOdd(int number) |
| { |
| while (isOddPrinted) |
| { |
| try |
| { |
| wait(); |
| } |
| catch (InterruptedException e) |
| { |
| e.printStackTrace(); |
| } |
| } |
| System.out.println(Thread.currentThread().getName()+" "+number); |
| isOddPrinted = true; |
| try |
| { |
| Thread.sleep(1000); |
| } |
| catch (InterruptedException e) |
| { |
| e.printStackTrace(); |
| } |
| notify(); |
| } |
|  |
| synchronized void printEven(int number) |
| { |
| while (! isOddPrinted) |
| { |
| try |
| { |
| wait(); |
| } |
| catch (InterruptedException e) |
| { |
| e.printStackTrace(); |
| } |
| } |
| System.out.println(Thread.currentThread().getName()+" "+number); |
| isOddPrinted = false; |
| try |
| { |
| T hread.sleep(1000); |
| } |
| catch (InterruptedException e) |
| { |
| e.printStackTrace(); |
| } |
| notify(); |
| } |
| } |
| public class Main |
| { |
| public static void main(String[] args) |
| { |
| sharedPrinter printer = new sharedPrinter(); |
| OddThread oddThread = new OddThread(20, printer); |
| oddThread.setName("—-pong"); |
| EvenThread evenThread = new EvenThread(20, printer); |
| evenThread.setName("ping — >"); |
| oddThread.start(); |
| evenThread.start(); |
| } |
| } |
|  |
|  |
|  |
| #Program 3 |
|  |
| #include <stdio.h> |
| bool find3Numbers(int A[], int arr\_size, int sum) |
| { |
| int l, r; |
| int t; |
| for int(t=0;t<l;t++) |
| { |
| for (int i = 0; i < arr\_size - 2; i++) { |
| for (int j = i + 1; j < arr\_size - 1; j++) { |
| for (int k = j + 1; k < arr\_size; k++) { |
| if (A[i] + A[j] + A[k] == sum) { |
| printf("%d, %d, %d", |
| A[i], A[j], A[k]); |
| return true; |
| } |
| } |
| } |
| } |
|  |
| return false; |
| } |
| int main() |
| { |
| int A[50]; |
| scanf("%d",A[i]); |
| int sum; |
| int arr\_size = sizeof(A) / sizeof(A[0]); |
| find3Numbers(A, arr\_size, sum); |
| return 0; |
| } |
|  |
|  |
|  |
| #Program 4 |
|  |
| #include <stdio.h> |
| int check\_anagram(char [], char []); |
| int main() |
| { |
| char a[100], b[100]; |
| printf("Enter two strings : \n"); |
| gets(a); |
| gets(b); |
|  |
| if (check\_anagram(a, b) == 1) |
| printf("%s and %s strings are anagrams\n",a,b); |
| else |
| printf("%s and %s strings are not anagrams\n"); |
|  |
| return 0; |
| } |
| int check\_anagram(char a[], char b[]) |
| { |
| int first[26] = {0}, second[26] = {0}, c=0; |
| while (a[c] != '\0') |
| { |
| first[a[c]-'a']++; |
| c++; |
| } |
| c = 0; |
| while (b[c] != '\0') |
| { |
| second[b[c]-'a']++; |
| c++; |
| } |
| for (c = 0; c < 26; c++) |
| { |
| if (first[c] != second[c]) |
| return 0; |
| } |
| return 1; |

}