

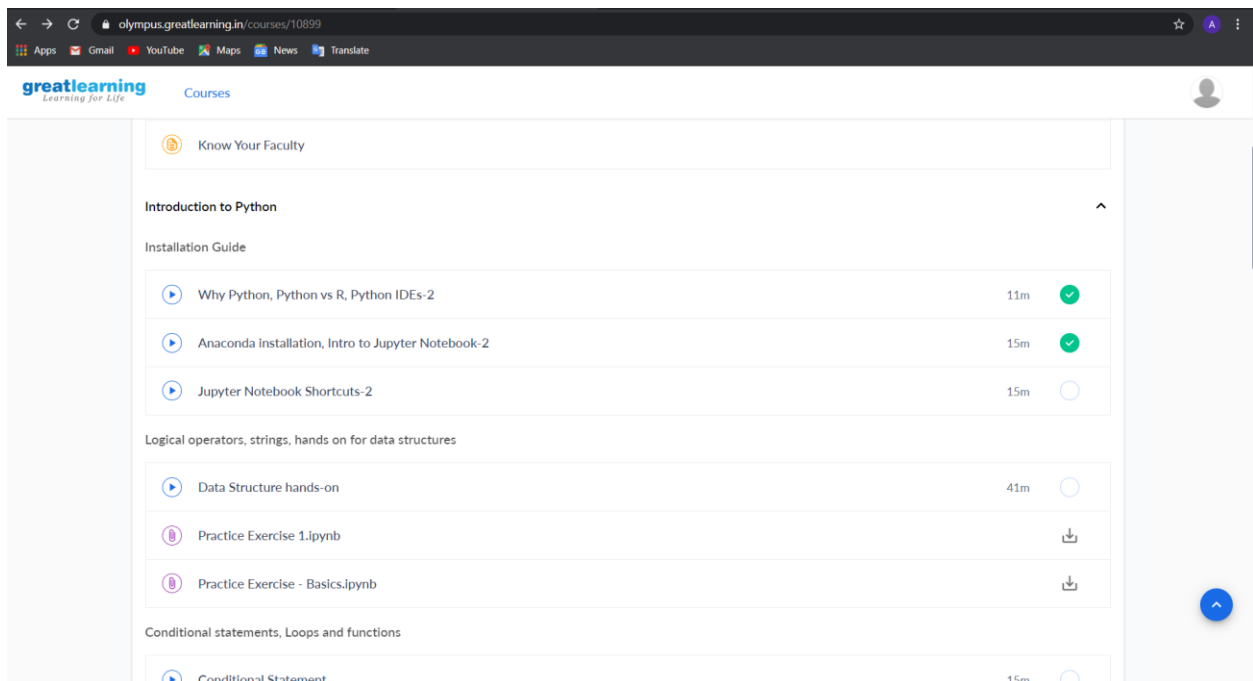
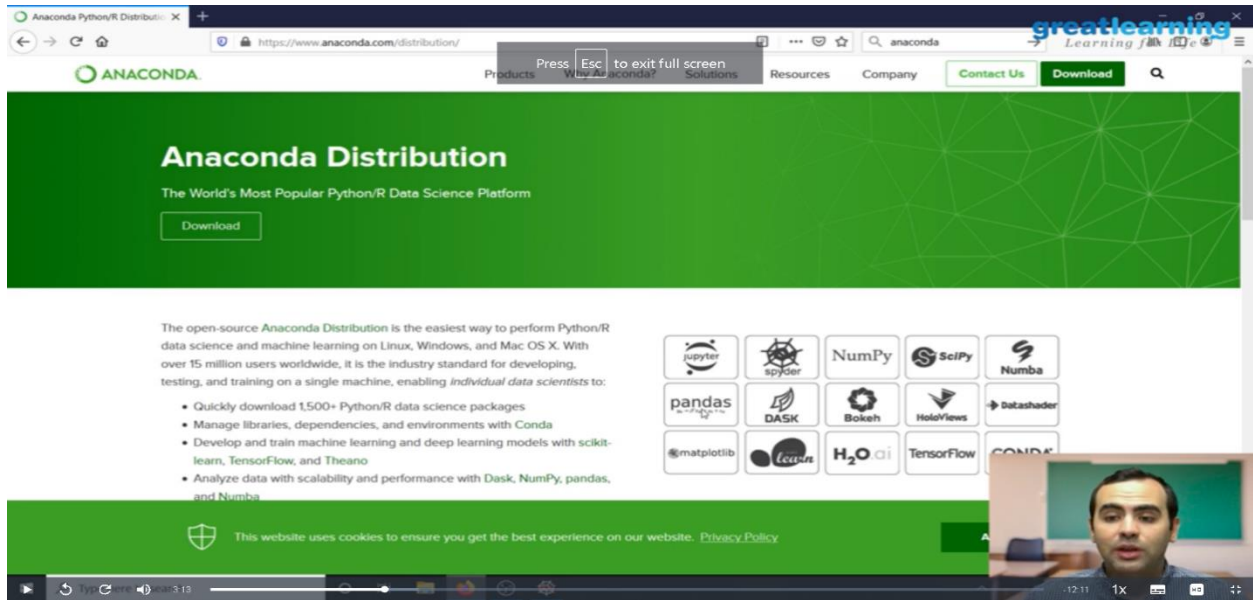
DAILY ONLINE ACTIVITIES SUMMARY

Date:	18-05-2020	Name:	Apoorva H P
Sem & Sec	VI A	USN:	4AL17CS011
Online Test Summary			
Subject	CNSC IA Test		
Max. Marks	60	Score	36
Certification Course Summary			
Course	Python for Machine learning		
Certificate Provider	GreatLearning	Duration	5hr
Coding Challenges			
Problem Statement: 1. Using methods charAt() & length() of String class, write a program to print the frequency of each character in a string. 2. Write down a java program to print even and odd numbers series respectively from two threads: t1 and t2 synchronizing on a shared object Let t1 print message "ping —>" and t2 print message ",—pong".			
Status: Completed, executed			
Uploaded the report in Github		Yes	
If yes Repository name		https://github.com/ashaapoorva/online-coding-and-certification-course	
Uploaded the report in slack		Yes	

Online Certification Details

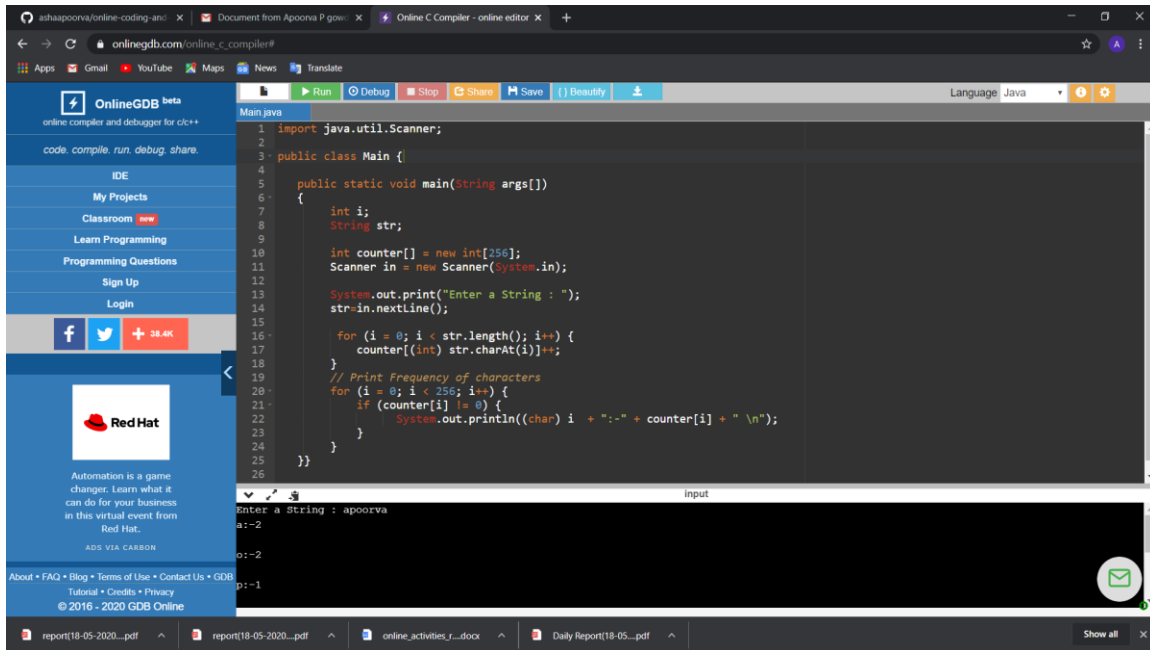
Modules completed:

- Why python, python vs R,
- Anaconda installation,Intro to Jupyter Notebook-2
- Jupyter Notebook Shortcut-2



Coding Challenge Details

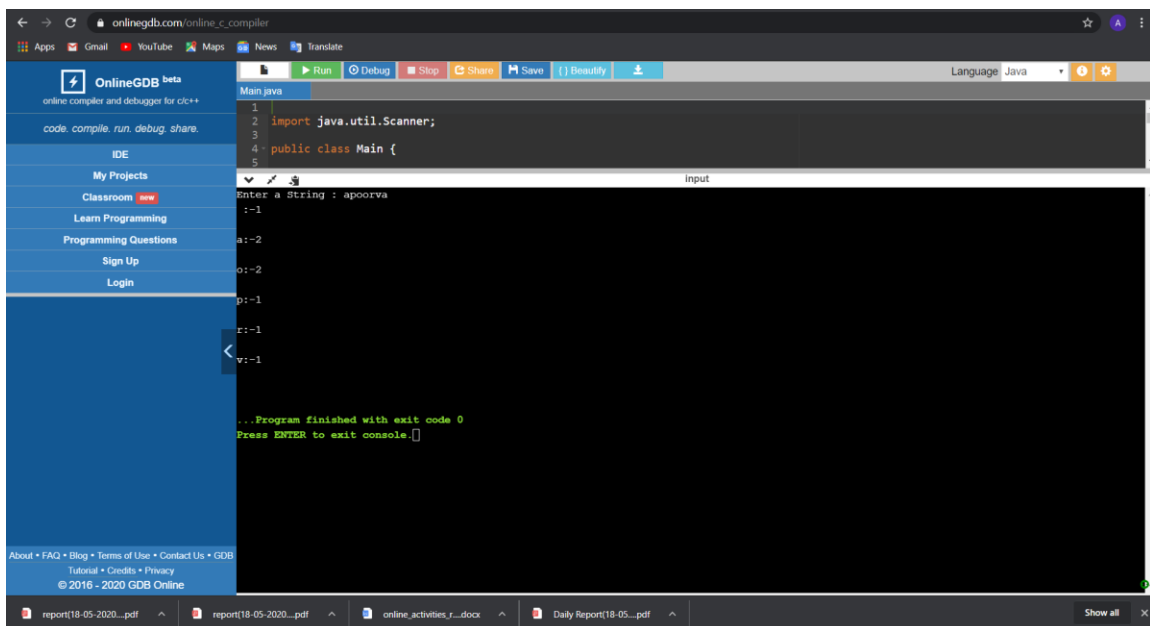
1. Using methods `charAt()` & `length()` of String class, write a program to print the frequency of each character in a string.



The screenshot shows the OnlineGDB IDE interface. The left sidebar contains navigation links: OnlineGDB beta, code compile, run, debug, share, IDE, My Projects, Classroom, Learn Programming, Programming Questions, Sign Up, and Login. The main editor displays a Java program that reads a string from the user and prints the frequency of each character. The program uses a Scanner to read input and a loop to iterate through the string, counting the frequency of each character using the `charAt()` method. The output shows the input string "apoorva" and the frequency of each character: a:2, p:1, o:2, r:1, v:1.

```
1 import java.util.Scanner;
2
3 public class Main {
4
5     public static void main(String args[])
6     {
7         int i;
8         String str;
9
10        int counter[] = new int[256];
11        Scanner in = new Scanner(System.in);
12
13        System.out.print("Enter a String : ");
14        str=in.nextLine();
15
16        for (i = 0; i < str.length(); i++) {
17            counter[(int) str.charAt(i)]++;
18        }
19        // Print Frequency of characters
20        for (i = 0; i < 256; i++) {
21            if (counter[i] != 0) {
22                System.out.println((char) i + ":" + counter[i] + " \n");
23            }
24        }
25    }
26 }
```

Input: Enter a String : apoorva
Output: a:2
p:1
o:2
r:1
v:1



The screenshot shows the OnlineGDB IDE interface with the same Java program as the previous screenshot. The program has been executed, and the output is displayed in the console. The output shows the input string "apoorva" and the frequency of each character: a:2, p:1, o:2, r:1, v:1. The program finished with exit code 0.

```
1 import java.util.Scanner;
2
3 public class Main {
4
5     public static void main(String args[])
6     {
7         int i;
8         String str;
9
10        int counter[] = new int[256];
11        Scanner in = new Scanner(System.in);
12
13        System.out.print("Enter a String : ");
14        str=in.nextLine();
15
16        for (i = 0; i < str.length(); i++) {
17            counter[(int) str.charAt(i)]++;
18        }
19        // Print Frequency of characters
20        for (i = 0; i < 256; i++) {
21            if (counter[i] != 0) {
22                System.out.println((char) i + ":" + counter[i] + " \n");
23            }
24        }
25    }
26 }
```

Input: Enter a String : apoorva
Output: a:2
p:1
o:2
r:1
v:1
...Program finished with exit code 0
Press ENTER to exit console.

2. Write down a java program to print even and odd numbers series respectively from two threads: t1 and t2 synchronizing on a shared object
Let t1 print message “ping —>” and t2 print message “,—pong”.

The screenshot displays the OnlineGDB IDE interface. The left sidebar contains navigation links such as 'My Projects', 'Classroom', 'Learn Programming', 'Programming Questions', 'Sign Up', and 'Login'. The main editor area shows a Java program with the following code:

```
1 class OddThread extends Thread
2 {
3     int limit;
4     sharedPrinter printer;
5     public OddThread(int limit, sharedPrinter printer)
6     {
7         this.limit = limit;
8         this.printer = printer;
9     }
10
11     @Override
12     public void run()
13     {
14         int oddNumber = 1;
15         while (oddNumber <= limit)
16         {
17             printer.printOdd(oddNumber);
18             oddNumber = oddNumber + 2;
19         }
20     }
21
22     class EvenThread extends Thread
23     {
24         int limit;
25         sharedPrinter printer;
26         public EvenThread(int limit, sharedPrinter printer)
27         {
28             this.limit = limit;
29             this.printer = printer;
30         }
31
32         @Override
33         public void run()
34         {
35             int evenNumber = 2;
36             while (evenNumber <= limit)
37             {
38                 printer.printEven(evenNumber);
39                 evenNumber = evenNumber + 2;
40             }
41         }
42     }
43
44     class sharedPrinter
45     {
46         boolean isOddPrinted = false;
47
48         synchronized void printOdd(int number)
49         {
50             while (isOddPrinted)
51             {
52                 try
53                 {
54                     wait();
55                 }
56                 catch (InterruptedException e)
57                 {
58                     e.printStackTrace();
59                 }
60             }
61             System.out.println(Thread.currentThread().getName()+" "+number);
62             isOddPrinted = true;
63             try
64             {
65                 Thread.sleep(1000);
66             }
67             catch (InterruptedException e)
68             {
69                 e.printStackTrace();
70             }
71         }
72
73         synchronized void printEven(int number)
74         {
75             while (!isOddPrinted)
76             {
77                 try
78                 {
79                     wait();
80                 }
81                 catch (InterruptedException e)
82                 {
83                     e.printStackTrace();
84                 }
85             }
86             System.out.println(Thread.currentThread().getName()+" "+number);
87             !isOddPrinted = false;
88             try
89             {
90                 Thread.sleep(1000);
91             }
92             catch (InterruptedException e)
93             {
94                 e.printStackTrace();
95             }
96         }
97     }
98 }
```

The code defines two classes, `OddThread` and `EvenThread`, which extend `Thread`. Both threads share a `sharedPrinter` class. The `sharedPrinter` class contains two synchronized methods, `printOdd` and `printEven`, which use `wait` and `sleep` to ensure that the threads print their respective numbers in an alternating sequence. The `printOdd` method prints the odd number and then sets `isOddPrinted` to `true`, while the `printEven` method prints the even number and then sets `!isOddPrinted` to `false`.

OnlineGDB beta
online compiler and debugger for c/c++
code compile run debug share

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login

Facebook Twitter + 28.4K

Red Hat
Automation is a game changer. Learn what it can do for your business in this virtual event from Red Hat.
ADS VIA CARBON

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2020 GDB Online

Main.java
67 catch (InterruptedException e)
68 {
69 e.printStackTrace();
70 }
71 notify();
72 }
73 }
74 synchronized void printEven(int number)
75 {
76 while (! isOddPrinted)
77 {
78 try
79 {
80 wait();
81 }
82 catch (InterruptedException e)
83 {
84 e.printStackTrace();
85 }
86 }
87 System.out.println(Thread.currentThread().getName()+" "+number);
88 isOddPrinted = false;
89 try
90 {
91 Thread.sleep(1000);
92 }
93 catch (InterruptedException e)
94 {
95 e.printStackTrace();
96 }
97 notify();
98 }
99 }

report18-05-2020...pdf report18-05-2020...pdf online_activities_r...docx Daily Report18-05...pdf Show all

OnlineGDB beta
online compiler and debugger for c/c++
code compile run debug share

IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login

Facebook Twitter + 28.4K

Red Hat
Automation is a game changer. Learn what it can do for your business in this virtual event from Red Hat.
ADS VIA CARBON

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2020 GDB Online

Main.java
80 wait();
81 }
82 catch (InterruptedException e)
83 {
84 e.printStackTrace();
85 }
86 }
87 System.out.println(Thread.currentThread().getName()+" "+number);
88 isOddPrinted = false;
89 try
90 {
91 Thread.sleep(1000);
92 }
93 catch (InterruptedException e)
94 {
95 e.printStackTrace();
96 }
97 notify();
98 }
99 }
100 public class Main
101 {
102 public static void main(String[] args)
103 {
104 sharedPrinter printer = new sharedPrinter();
105 OddThread oddThread = new OddThread(20, printer);
106 oddThread.setName("--pong");
107 EvenThread evenThread = new EvenThread(20, printer);
108 evenThread.setName("ping ->");
109 oddThread.start();
110 evenThread.start();
111 }
112 }
113 }

report18-05-2020...pdf report18-05-2020...pdf online_activities_r...docx Daily Report18-05...pdf Show all

The screenshot shows the OnlineGDB IDE interface. The left sidebar contains navigation links: IDE, My Projects, Classroom (marked 'new'), Learn Programming, Programming Questions, Sign Up, and Login. The main editor area displays a Java file named 'Main.java' with the following code:

```
1 class OddThread extends Thread
2 {
3     int limit;
4     sharedPrinter printer;
5 }
6
7 pong 1
8 ping -> 2
9 pong 3
10 ping -> 4
11 pong 5
12 ping -> 6
13 pong 7
14 ping -> 8
15 pong 9
16 ping -> 10
17 pong 11
18 ping -> 12
19 pong 13
20 ping -> 14
21 pong 15
22 ping -> 16
23 pong 17
24 ping -> 18
25 pong 19
26 ping -> 20
27
28 ...Program finished with exit code 0
29 Press ENTER to exit console.
```

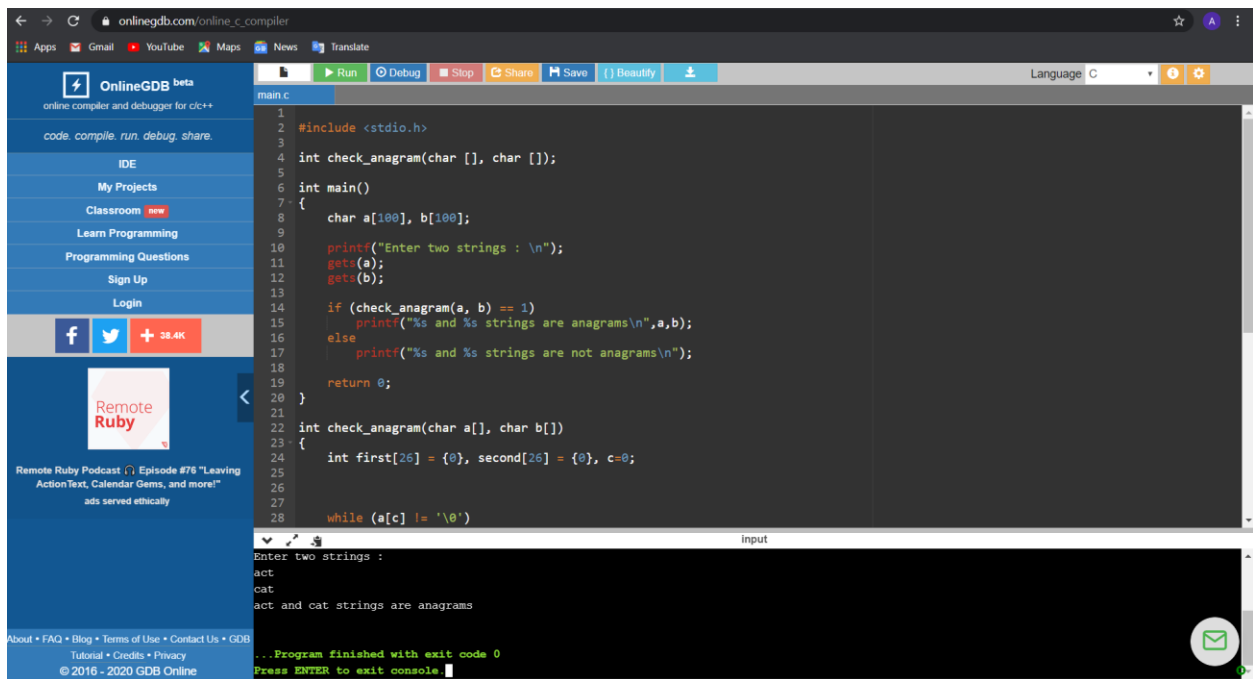
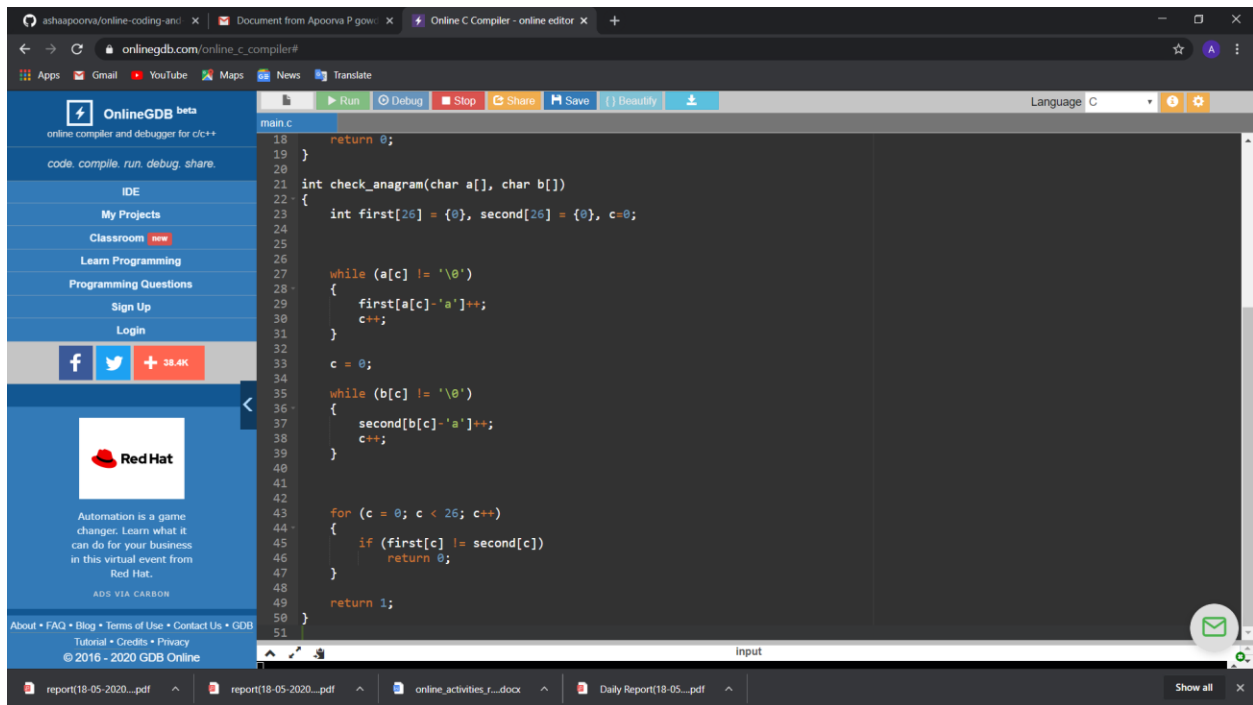
The bottom of the browser shows several open tabs, including 'report(18-05-2020...pdf' and 'online_activities_f...docx'.

3.write a c programme to check whether two strings are anagram or not

The screenshot shows the OnlineGDB IDE interface with the language set to C. The left sidebar is similar to the first image but includes a Red Hat advertisement. The main editor area displays a C file named 'main.c' with the following code:

```
1 #include <stdio.h>
2
3 int check_anagram(char [], char []);
4
5 int main()
6 {
7     char a[100], b[100];
8
9     printf("Enter two strings : \n");
10    gets(a);
11    gets(b);
12
13    if (check_anagram(a, b) == 1)
14        printf("%s and %s strings are anagrams\n",a,b);
15    else
16        printf("%s and %s strings are not anagrams\n");
17
18    return 0;
19 }
20
21 int check_anagram(char a[], char b[])
22 {
23     int first[26] = {0}, second[26] = {0}, c=0;
24
25
26
27     while (a[c] != '\0')
28     {
29         first[a[c]-'a']++;
30         c++;
31     }
32
33     c = 0;
34 }
```

The bottom of the browser shows several open tabs, including 'report(18-05-2020...pdf' and 'online_activities_f...docx'.



Coding Test Details

The screenshot shows a web browser window with two tabs: 'Cryptography Network Security' and 'Largest Tech Community | Hack...'. The address bar displays the URL 'techgig.com/challenge/result/cnsc-i-a-1/MOJCK20zaFZMZVNpZ0pBOXFoa38SUT09'. Below the browser window, a white card displays the test results under the 'Results' tab. The card is divided into four sections, each representing a submitted test. Each section includes a green checkmark icon, the test name, the challenge identifier 'CNSC I A 1', and the score. The scores are: Test 4 (8/28), Test 1 (6/8), Test 2 (10/12), and Test 3 (12/12).

Test	Score	Max Score
Test 4 submitted	8	28
Test 1 submitted	6	8
Test 2 submitted	10	12
Test 3 submitted	12	12