DAILY ONLINE ACTIVITIES SUMMARY

Date:	**23-05-2020**	**Name:**	**D Jasmine Joyline**
Sem & Sec	**VI A**	**USN:**	**4AL17CS024**

Online Test Summary

Subject	**Python Application Programming**		
Max. Marks	**30**	**Score**	**21**

Certification Course Summary

Course	**Python Bootcamp 2020: Build 15 working Applications and Games**		
Certificate Provider	**Udemy**	**Duration**	**32hr**

Coding Challenges

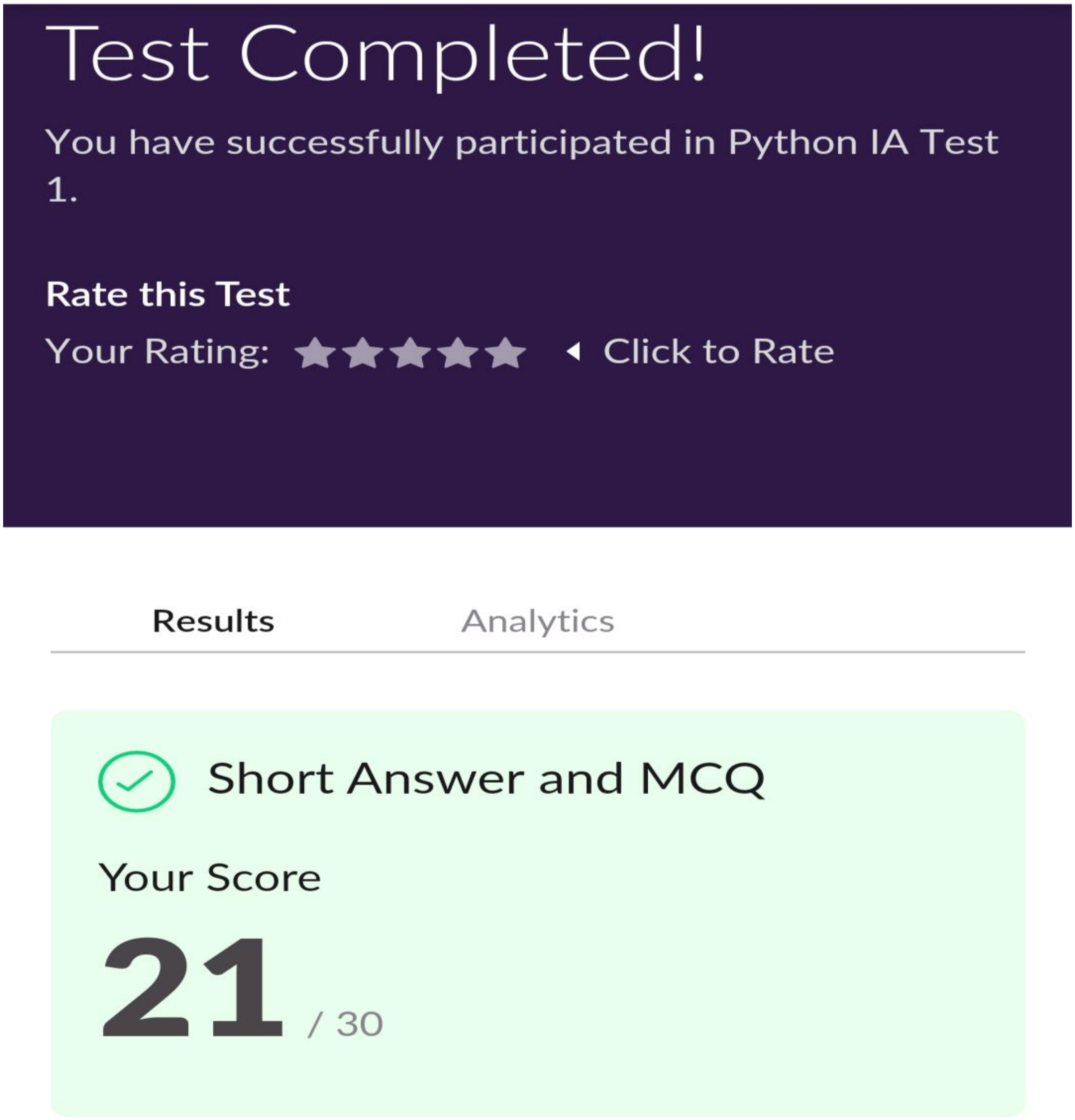
Problem Statement:
1.C program to print the triangular Number Series
2.Rotten oranges problem

Status:Completed

Uploaded the report in Github	**Yes**
If yes Repository name	https://github.com/alvas-education-foundation/D_Jasmine_Joyline/tree/master/daily_progress
Uploaded the report in slack	**Yes**

Online Test Details:

PAP IA TEST



Test Completed!

You have successfully participated in Python IA Test 1.

Rate this Test

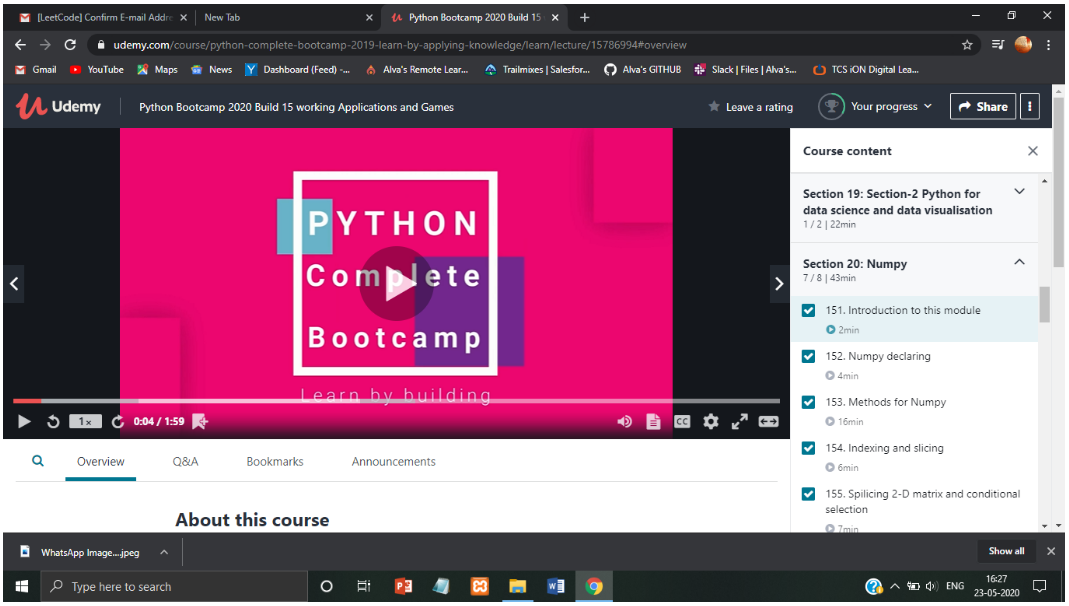
Your Rating: ★★★★★

Short Answer and MCQ

Your Score

21 / 30

Certification Course Details:



udemy.com/course/python-complete-bootcamp-2019-learn-by-applying-knowledge/learn/lecture/15786994#overview

Python Bootcamp 2020 Build 15 working Applications and Games

PYTHON Complete Bootcamp

Learn by building

About this course

Course content

Section 19: Section-2 Python for data science and data visualisation
1 / 2 | 22min

Section 20: Numpy
7 / 8 | 43min

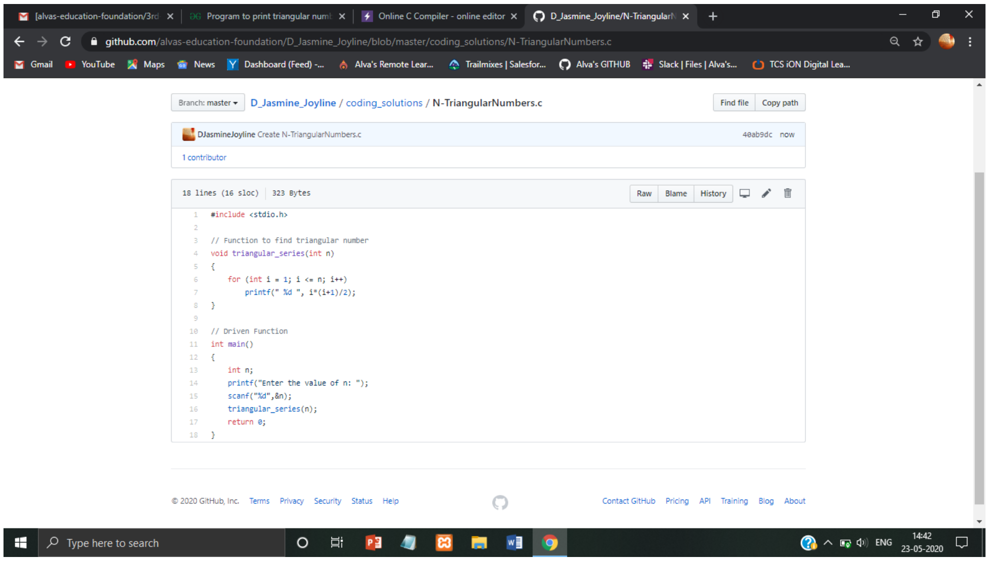
- ☑ 151. Introduction to this module — 2min
- ☑ 152. Numpy declaring — 4min
- ☑ 153. Methods for Numpy — 16min
- ☑ 154. Indexing and slicing — 6min
- ☑ 155. Spilicing 2-D matrix and conditional selection — 7min

Modules that I have covered today:

- Numpy
- Project3: Hangman game
- Project4:Tic Tac Toe

Coding Challenges Details

1.Write a C program to print the Triangle Number Series


Program to print triangular numb
Online C Compiler - online editor
D_Jasmine_Joyline/N-TriangularN
Branch: master
Find file
Copy path
Raw
Blame
History
Type here to search
ENG
14:42
23-05-2020

github.com/alvas-education-foundation/D_Jasmine_Joyline/blob/master/coding_solutions/N-TriangularNumbers.c

D_Jasmine_Joyline / coding_solutions / N-TriangularNumbers.c

DJasmineJoyline Create N-TriangularNumbers.c 40ab9dc now

1 contributor

18 lines (16 sloc) | 323 Bytes

```c
#include <stdio.h>

// Function to find triangular number
void triangular_series(int n)
{
    for (int i = 1; i <= n; i++)
        printf(" %d ", i*(i+1)/2);
}

// Driven Function
int main()
{
    int n;
    printf("Enter the value of n: ");
    scanf("%d",&n);
    triangular_series(n);
    return 0;
}
```

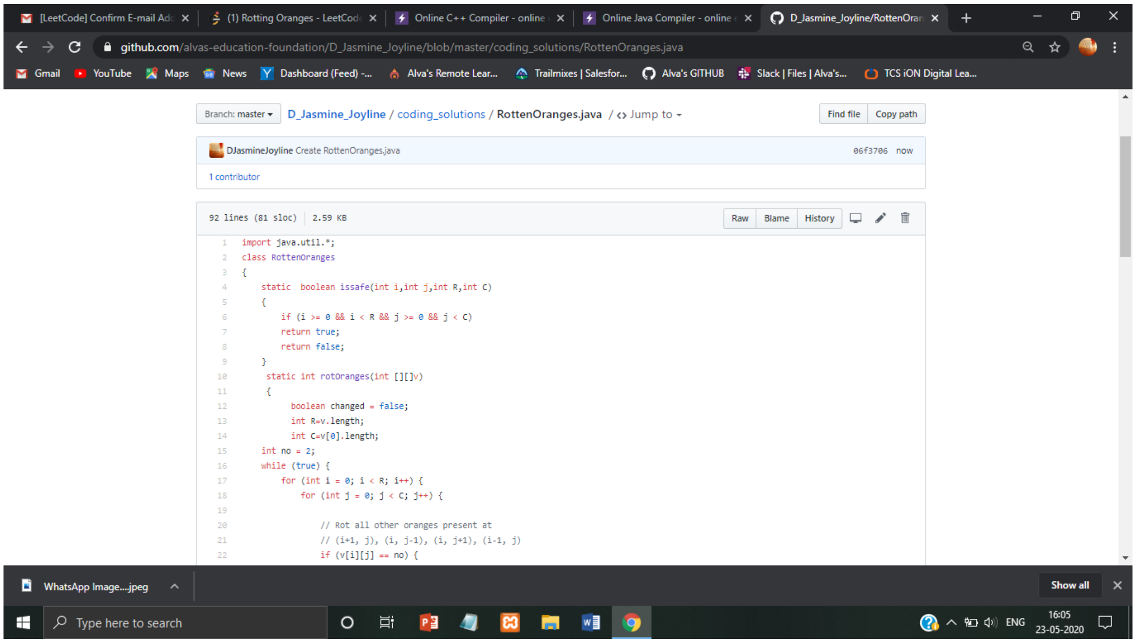
© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

2. In a given grid, each cell can have one of three values:

- the value `0` representing an empty cell;
- the value `1` representing a fresh orange;
- the value `2` representing a rotten orange.

Every minute, any fresh orange that is adjacent (4-directionally) to a rotten orange becomes rotten.

Return the minimum number of minutes that must elapse until no cell has a fresh orange. If this is impossible, return `-1` instead.


[LeetCode] Confirm E-mail Add
(1) Rotting Oranges - LeetCode
Online C++ Compiler - online
Online Java Compiler - online
D_Jasmine_Joyline/RottenOran
github.com/alvas-education-foundation/D_Jasmine_Joyline/blob/master/coding_solutions/RottenOranges.java
Gmail
YouTube
Maps
News
Dashboard (Feed) -...
Alva's Remote Lear...
Trailmixes | Salesfor...
Alva's GITHUB
Slack | Files | Alva's...
TCS iON Digital Lea...
Branch: master
D_Jasmine_Joyline / coding_solutions / RottenOranges.java / <> Jump to
Find file
Copy path
DJasmineJoyline Create RottenOranges.java
06f3706 now
1 contributor
92 lines (81 sloc) | 2.59 KB
Raw
Blame
History
WhatsApp Image....jpeg
Show all
Type here to search
ENG
16:05
23-05-2020

```java
import java.util.*;
class RottenOranges
{
    static  boolean issafe(int i,int j,int R,int C)
    {
        if (i >= 0 && i < R && j >= 0 && j < C)
        return true;
        return false;
    }
     static int rotOranges(int [][]v)
     {
          boolean changed = false;
          int R=v.length;
          int C=v[0].length;
    int no = 2;
    while (true) {
        for (int i = 0; i < R; i++) {
            for (int j = 0; j < C; j++) {

                // Rot all other oranges present at
                // (i+1, j), (i, j-1), (i, j+1), (i-1, j)
                if (v[i][j] == no) {
```