LAB SESSION 05:

Date of the Session:___/___/____ **Time of the Session:_____to_______**

Pre-Lab:

1) Explain why 0-1 Knapsack problems cannot be solved using greedy method unlike fractional knapsack.

Solution

In 0-1 Knapsack, items cannot be broken which means the thief should take the item as a whole or should leave it. This is reason behind calling it as 0-1 Knapsack. Hence, in case of 0-1 Knapsack, the value of xi can be either 0 or 1, where other constraints remain the same. 0-1 Knapsack cannot be solved by Greedy approach. Greedy approach does not ensure an optimal solution. In many instances, Greedy approach may give an optimal solution.

2) Categorize the Following as single source or multiple source shortest path algorithms.

Floyd-Warshall algorithm –
Dijkstra's algorithm –
Bellman-Ford algorithm –

Solution

Floyd-Warshall algorithm – Multiple Source Shortest Path
Dijkstra's algorithm – Single Source Shortest Path
Bellman-Ford algorithm – Single Source Shortest Path

3) List down various shortest path greedy algorithms.

Solution

The shortest path problem is about finding a path between vertices in a graph such that the total sum of the edges weights is minimum. This problem could be solved easily using (BFS) if all edge weights were (), but here weights can take any value. Three different algorithms are discussed below depending on the use-case.

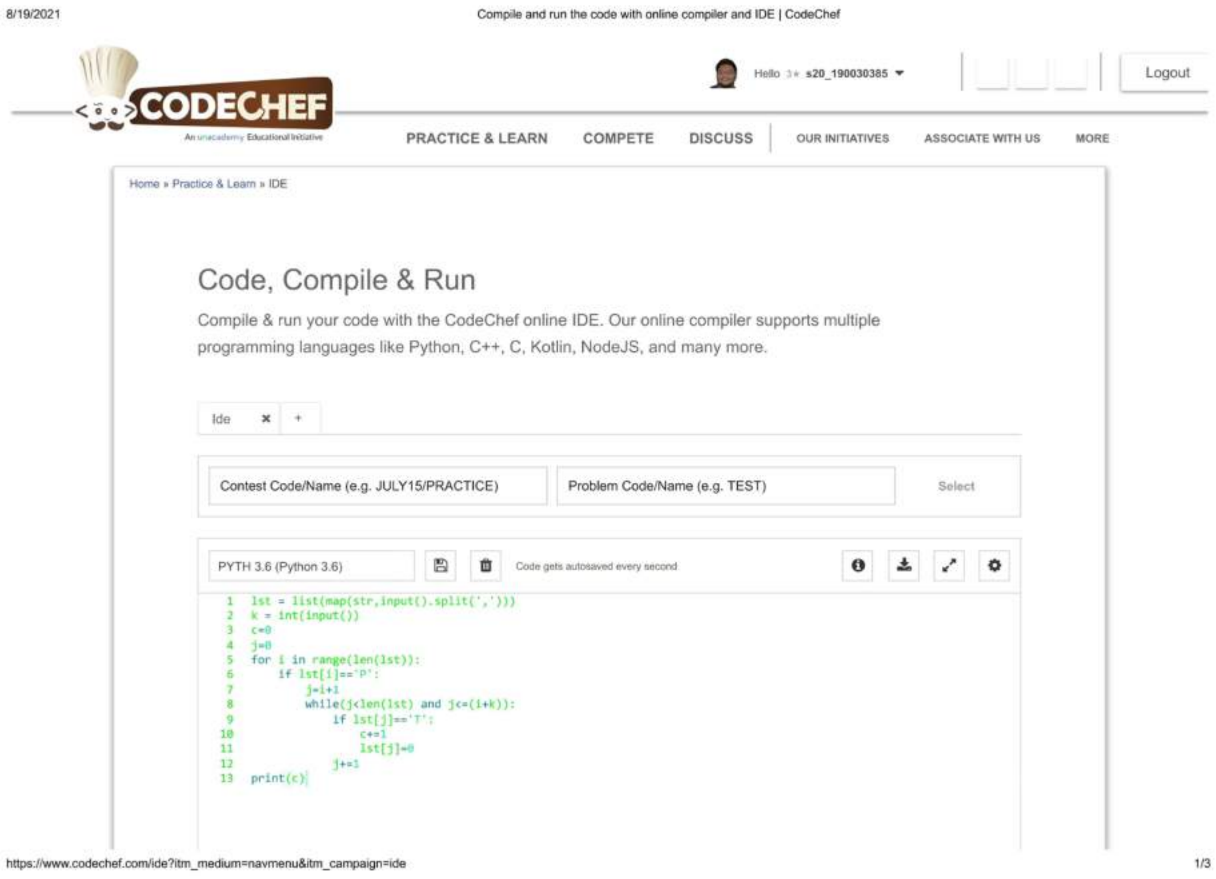
Floyd-Warshall algorithm – Multiple Source Shortest Path
Dijkstra's algorithm – Single Source Shortest Path
Bellman-Ford algorithm – Single Source Shortest Path

In-Lab:

1) Given an array of size n that has the following specifications:
   a. Each element in the array contains either a police officer or a thief.
   b. Each police officer can catch only one thief.
   c. A police officer cannot catch a thief who is more than K units away from the police officer.

We need to find the maximum number of thieves that can be caught.

Input
arr [] = {'P', 'T', 'T', 'P', 'T'},
k = 1.
Output
2
Here maximum 2 thieves can be caught; first police officer catches first thief and second police officer can catch either second or third thief.


8/19/2021 Compile and run the code with online compiler and IDE | CodeChef

Code, Compile & Run

Compile & run your code with the CodeChef online IDE. Our online compiler supports multiple programming languages like Python, C++, C, Kotlin, NodeJS, and many more.

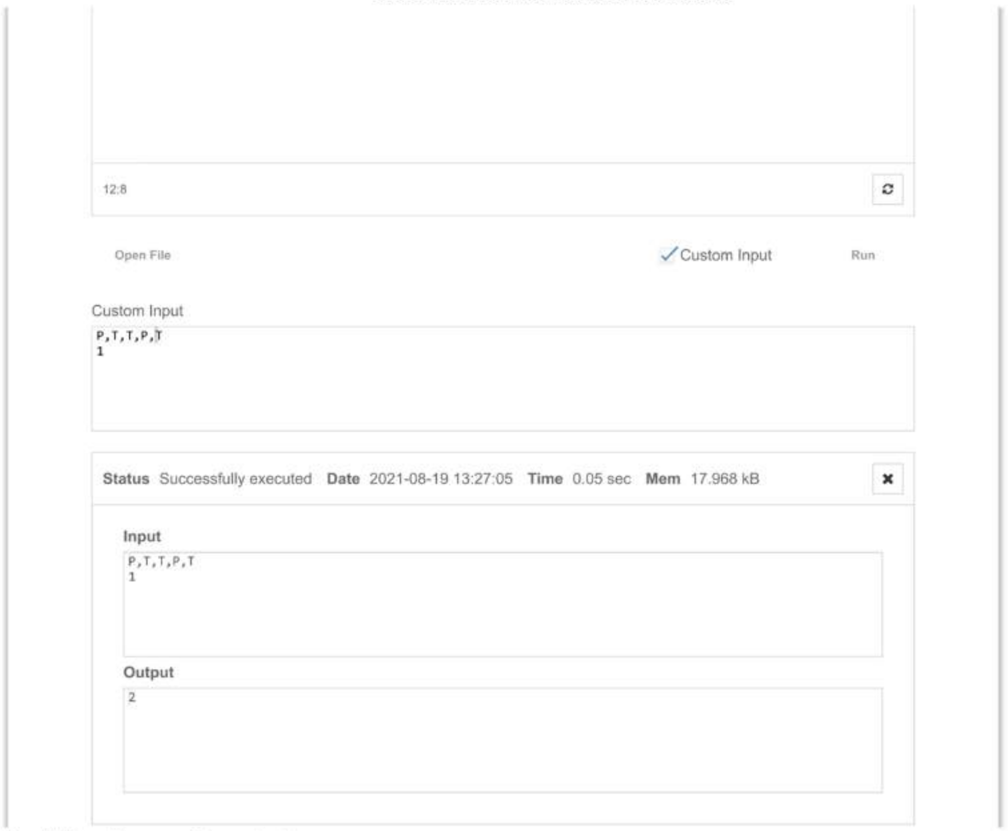
Ide
Contest Code/Name (e.g. JULY15/PRACTICE)
Problem Code/Name (e.g. TEST)
Select

PYTH 3.6 (Python 3.6)

Code gets autosaved every second

```
lst = list(map(str,input().split(',')))
k = int(input())
c=0
j=0
for i in range(len(lst)):
    if lst[i]=='P':
        j=i+1
        while(j<len(lst) and j<=(i+k)):
            if lst[j]=='T':
                c+=1
                lst[j]=0
            j+=1
print(c)
```

https://www.codechef.com/ide?itm_medium=navmenu&itm_campaign=ide 1/3


12:8

Open File ✓Custom Input Run

Custom Input

```
P,T,T,P,T
1
```

Status Successfully executed Date 2021-08-19 13:27:05 Time 0.05 sec Mem 17.968 kB

Input

```
P,T,T,P,T
1
```

Output

```
2
```

2) Given n non-negative integers a1, a2, ..., an, where each represents a point at coordinate (i, ai). n vertical lines are drawn such that the two endpoints of the line i is at (i, ai) and (i, 0). Find two lines, which, together with the x-axis forms a container, such that the container contains the most water. Notice that you may not slant the container.

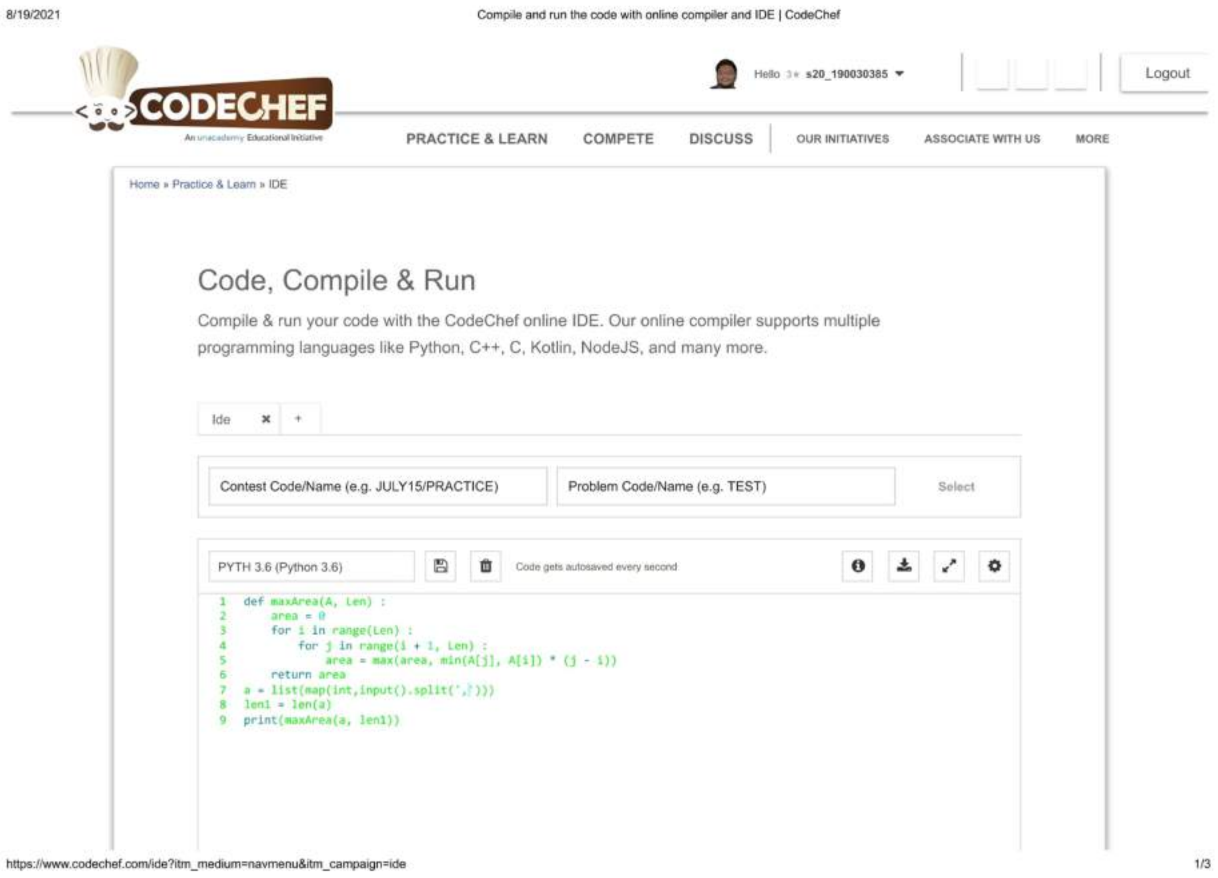
   Input

   height = [1,8,6,2,5,4,8,3,7]

   Output

   49

   Explanation: The above vertical lines are represented by array [1,8,6,2,5,4,8,3,7]. In this case, the max area of water (blue section) the container


8/19/2021
Compile and run the code with online compiler and IDE | CodeChef
CODECHEF
Hello s20_190030385
Logout
PRACTICE & LEARN
COMPETE
DISCUSS
OUR INITIATIVES
ASSOCIATE WITH US
MORE
Home » Practice & Learn » IDE
Ide
Select
https://www.codechef.com/ide?itm_medium=navmenu&itm_campaign=ide
1/3

Code, Compile & Run

Compile & run your code with the CodeChef online IDE. Our online compiler supports multiple programming languages like Python, C++, C, Kotlin, NodeJS, and many more.

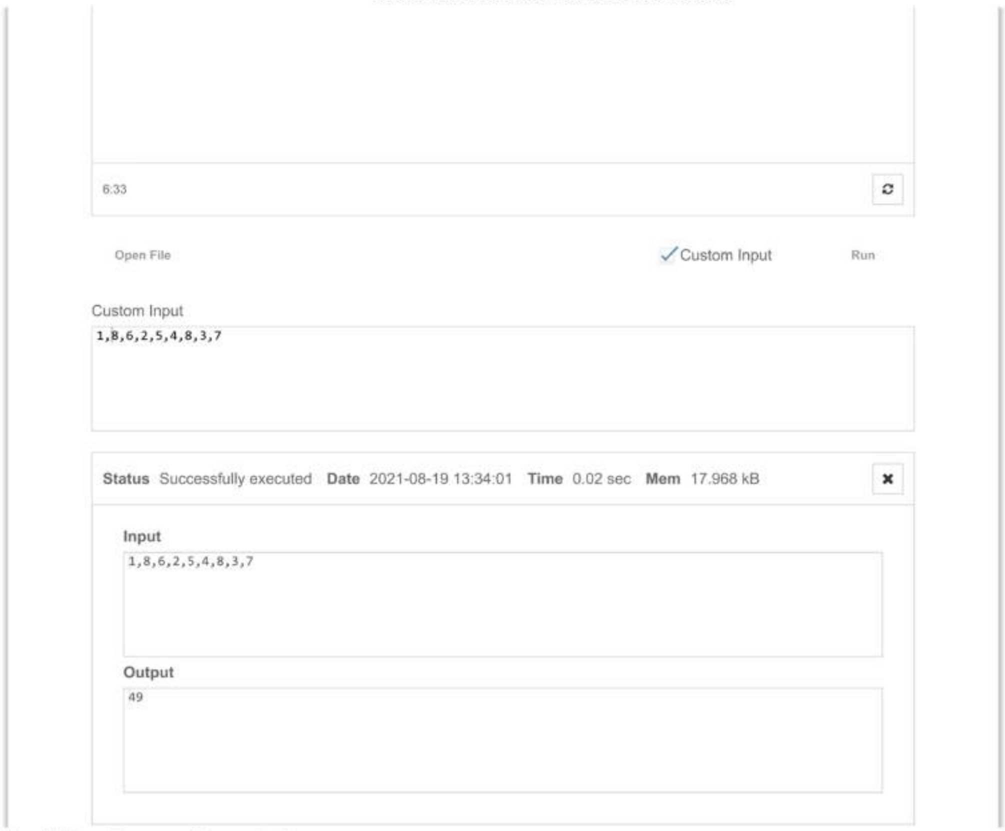
Contest Code/Name (e.g. JULY15/PRACTICE)

Problem Code/Name (e.g. TEST)

PYTH 3.6 (Python 3.6)

Code gets autosaved every second

```
def maxArea(A, Len) :
    area = 0
    for i in range(Len) :
        for j in range(i + 1, Len) :
            area = max(area, min(A[j], A[i]) * (j - i))
    return area
a = list(map(int,input().split(',')))
len1 = len(a)
print(maxArea(a, len1))
```


6:33

Open File ✓Custom Input Run

Custom Input

```
1,8,6,2,5,4,8,3,7
```

Status Successfully executed **Date** 2021-08-19 13:34:01 **Time** 0.02 sec **Mem** 17.968 kB

Input

```
1,8,6,2,5,4,8,3,7
```

Output

```
49
```

Post-Lab:

1) Given an array of jobs where every job has a deadline and associated profit if the job is finished before the deadline. It is also given that every job takes a single unit of time, so the minimum possible deadline for any job is 1. How to maximize total profit if only one job can be scheduled at a time.

Input

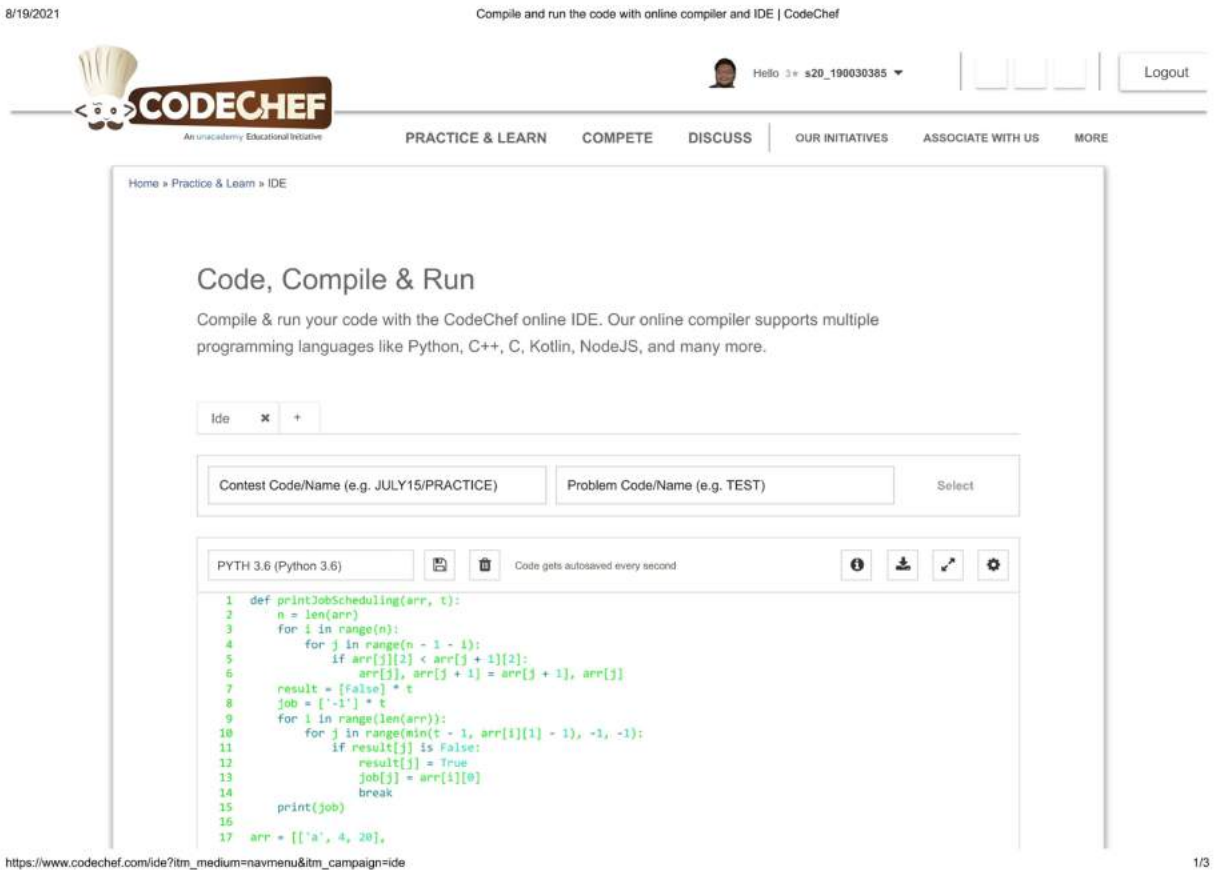
4

Job ID	Deadline	Profit
a	4	20
b	1	10
c	1	40
d	1	30

Output

60

profit sequence of jobs is c, a



8/19/2021 Compile and run the code with online compiler and IDE | CodeChef

CODECHEF
An unacademy Educational Initiative
Hello s20_190030385
Logout
PRACTICE & LEARN
COMPETE
DISCUSS
OUR INITIATIVES
ASSOCIATE WITH US
MORE
Home » Practice & Learn » IDE

Code, Compile & Run

Compile & run your code with the CodeChef online IDE. Our online compiler supports multiple programming languages like Python, C++, C, Kotlin, NodeJS, and many more.

Ide
Contest Code/Name (e.g. JULY15/PRACTICE)
Problem Code/Name (e.g. TEST)
Select

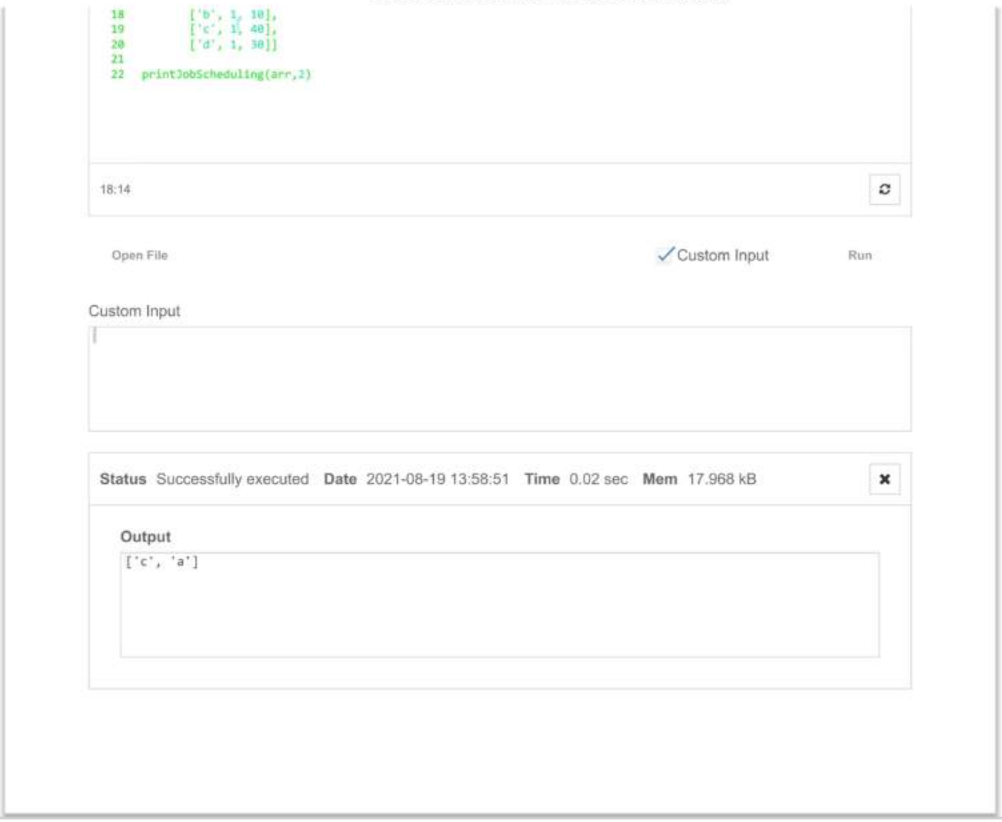
PYTH 3.6 (Python 3.6)

Code gets autosaved every second

```python
def printJobScheduling(arr, t):
    n = len(arr)
    for i in range(n):
        for j in range(n - 1 - i):
            if arr[j][2] < arr[j + 1][2]:
                arr[j], arr[j + 1] = arr[j + 1], arr[j]
    result = [False] * t
    job = ['-1'] * t
    for i in range(len(arr)):
        for j in range(min(t - 1, arr[i][1] - 1), -1, -1):
            if result[j] is False:
                result[j] = True
                job[j] = arr[i][0]
                break
    print(job)

arr = [['a', 4, 20],
```

https://www.codechef.com/ide?itm_medium=navmenu&itm_campaign=ide 1/3



```
        ['b', 1, 10],
        ['c', 1, 40],
        ['d', 1, 30]]

printJobScheduling(arr,2)
```

18:14

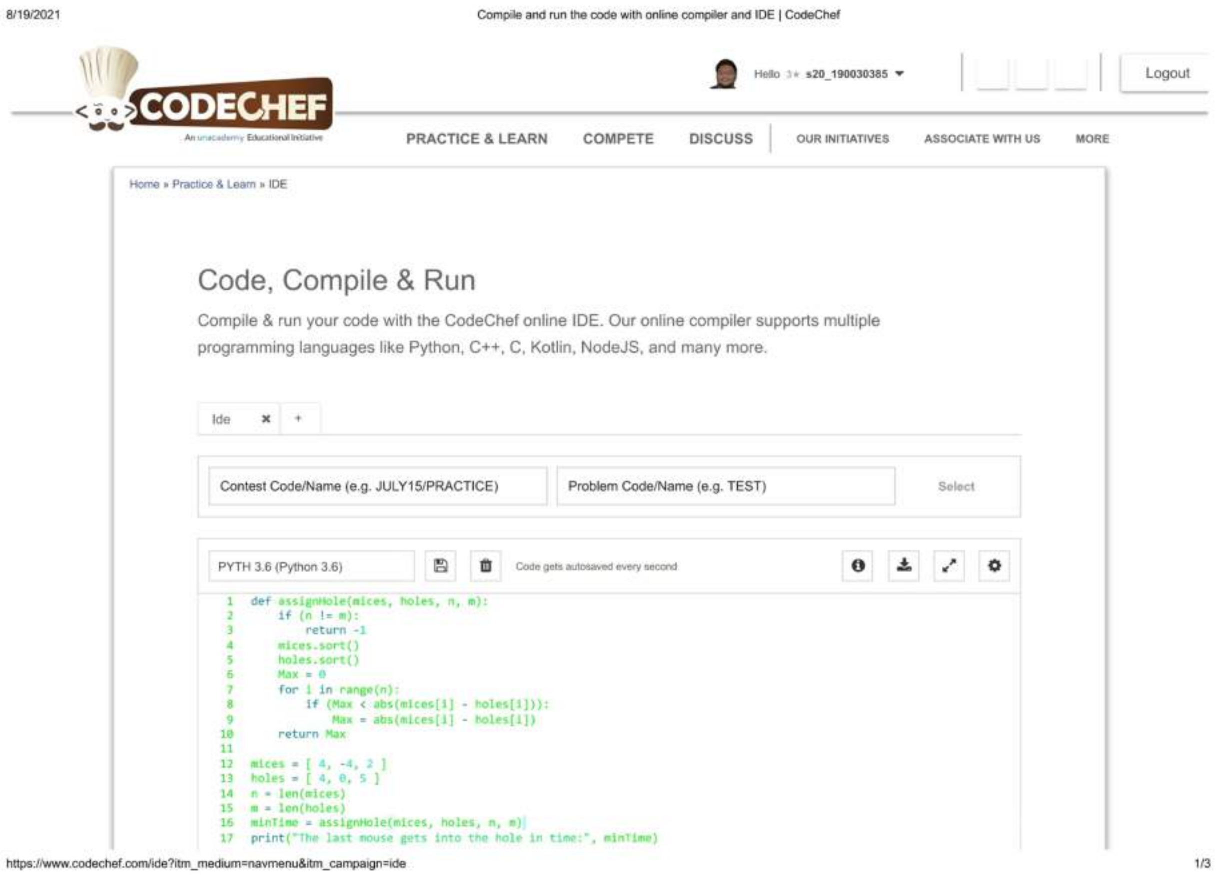
Status Successfully executed **Date** 2021-08-19 13:58:51 **Time** 0.02 sec **Mem** 17.968 kB

Output

```
['c', 'a']
```

2) There are N Mice and N holes are placed in a straight line. Each hole can accommodate only 1 mouse. A mouse can stay at his position, move one step right from x to x + 1, or move one step left from x to x − 1. Any of these moves consumes 1 minute.Assign mice to holes so that the time when the last mouse gets inside a hole is minimized.

Example: positions of mice are: 4 -4 2
Positions of holes are: 4 0 5
Input:
A: list of positions of mice
B: list of positions of holes
Output:
single integer value


8/19/2021 Compile and run the code with online compiler and IDE | CodeChef

CODECHEF
An unacademy Educational Initiative
Hello s20_190030385
Logout
PRACTICE & LEARN
COMPETE
DISCUSS
OUR INITIATIVES
ASSOCIATE WITH US
MORE
Home » Practice & Learn » IDE

Code, Compile & Run

Compile & run your code with the CodeChef online IDE. Our online compiler supports multiple programming languages like Python, C++, C, Kotlin, NodeJS, and many more.

Ide
Contest Code/Name (e.g. JULY15/PRACTICE)
Problem Code/Name (e.g. TEST)
Select

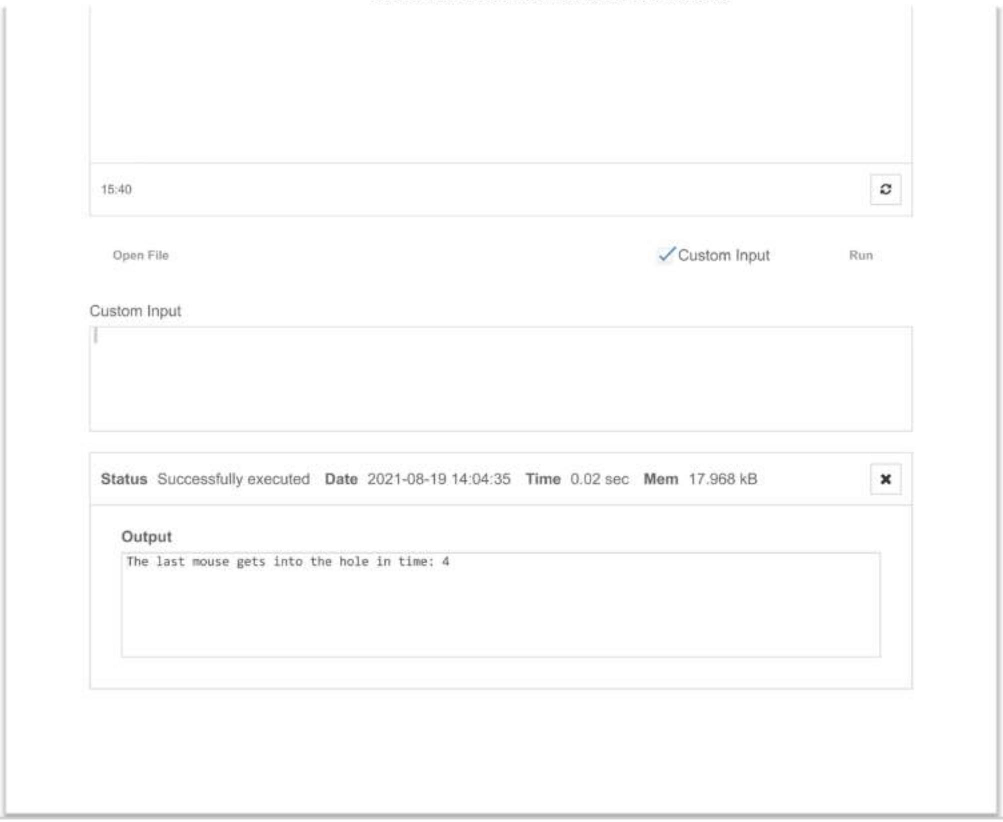
PYTH 3.6 (Python 3.6)

Code gets autosaved every second

```python
def assignHole(mices, holes, n, m):
    if (n != m):
        return -1
    mices.sort()
    holes.sort()
    Max = 0
    for i in range(n):
        if (Max < abs(mices[i] - holes[i])):
            Max = abs(mices[i] - holes[i])
    return Max

mices = [ 4, -4, 2 ]
holes = [ 4, 0, 5 ]
n = len(mices)
m = len(holes)
minTime = assignHole(mices, holes, n, m)
print("The last mouse gets into the hole in time:", minTime)
```

https://www.codechef.com/ide?itm_medium=navmenu&itm_campaign=ide 1/3



Status Successfully executed Date 2021-08-19 14:04:35 Time 0.02 sec Mem 17.968 kB

Output

```
The last mouse gets into the hole in time: 4
```

(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured:________out of______ Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation: