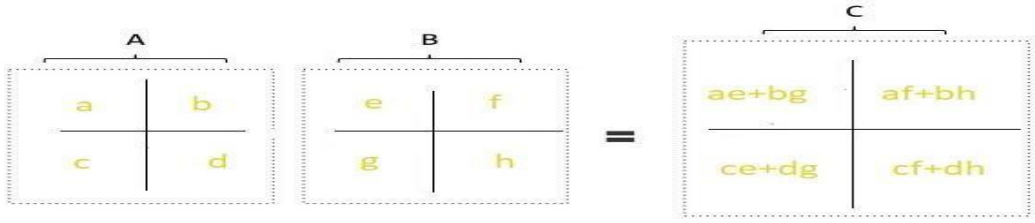
190031594 T.Lalitha Vyshnavi

LAB SESSION 04:

Date of the Session: __/___/_____ **Time of the Session: ____to_____**

Pre-Lab:

1) Trace the output of the following matrix multiplication using Strassen's Multiplication Method



A, B and C are the Matrices of Size NxN
a, b, c and d are the sub-Matrices of A of size N/2xN/2
e, f, g and h are the sub-Matrices of B of size N/2xN/2

1900303219

Strassen's Algorithm for Matrix Multiplication (for 2x2 matrix)

$$M_1 = (A_{1,1} + A_{2,2})\ (B_{1,1} + B_{2,2})$$

$$M_2 = (A_{2,1} + A_{2,2})\ (B_{1,1})$$

$$M_3 = A_{1,1}\ (B_{1,2} - B_{2,2})$$

$$M_4 = A_{2,2}\ (B_{2,1} - B_{1,1})$$

$$M_5 = (A_{1,1} + A_{1,2})\ B_{2,2}$$

$$M_6 = (A_{2,1} - A_{1,1})\ (B_{1,1} + B_{1,2})$$

$$M_7 = (A_{1,2} - A_{2,2})\ (B_{2,1} + B_{2,2})$$

$$C_{1,1} = M_1 + M_4 - M_5 + M_7$$

$$C_{1,2} = M_3 + M_5$$

$$C_{2,1} = M_2 + M_4$$

$$C_{2,2} = M_1 - M_2 + M_3 + M_6$$

1900030319

Steps:

1. First find M_{ic} values for the given matrix using above defined formula

where

$$k = n^3 - 1$$

2) After finding M_{ic} values substitute them in the resultant matrix cells using the formulas stated using addition and subtraction operations on M_{ic} values.

3) we will get our desired output.

2) China had recently banned cryptocurrency. Due to this cryptocurrency cost has fallen drastically. Mr. Lee has lost a lot so he could not afford a stock advisor. He came to you (his friend) for help. He has the prices of stock on i'th day. Help him to find the maximum profit he can achieve. You may complete as many transactions as you like.

Input

7

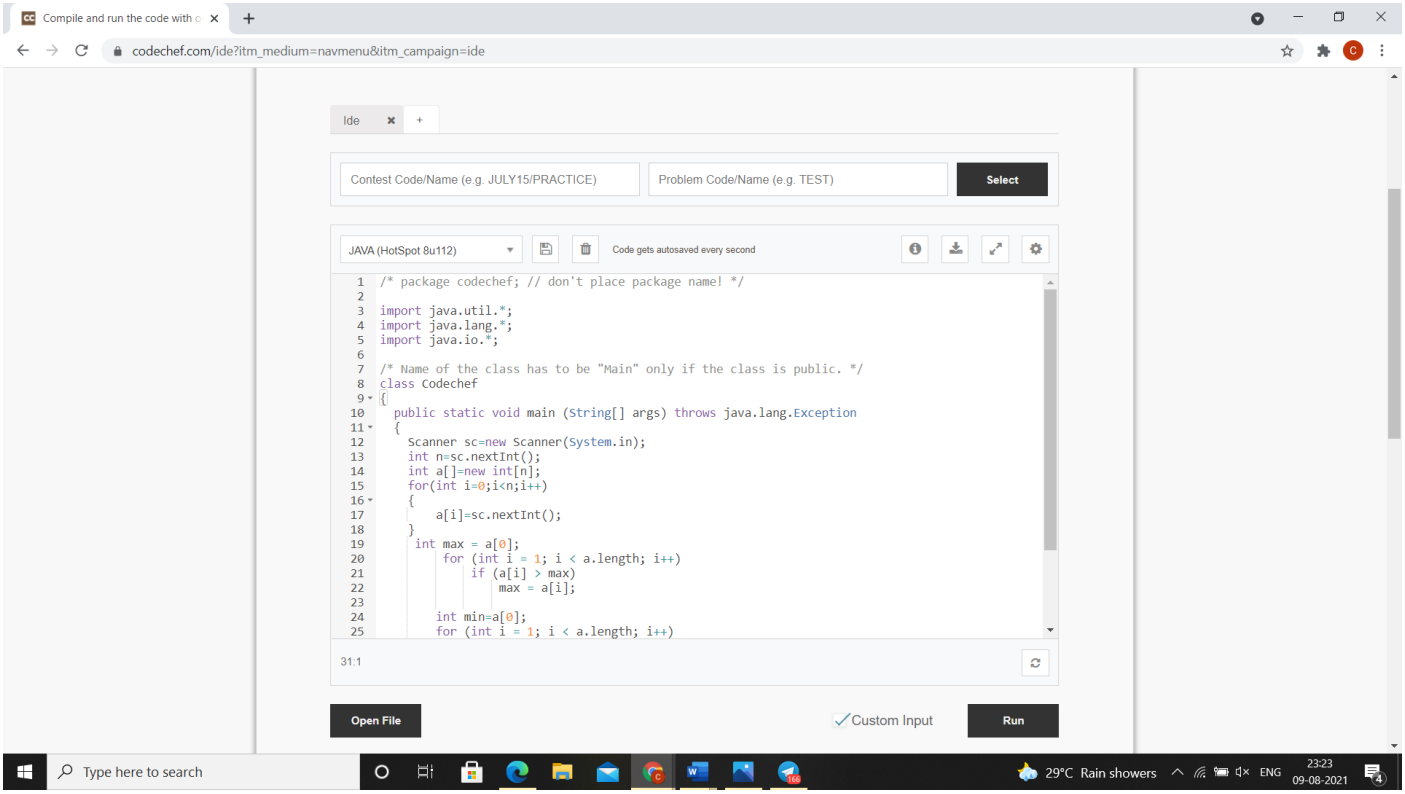
[1,2,3,4,5,6,7]

Output

6

Explanation:

Buy the stock on 1st day at cost 1Rupee and sell the stock on 7th day at cost 7 Rupee and get profit of 6 rupees.



In-Lab:

1) You are given 2 matrices of any size N*N. Write a program to find the product of the two matrixes (use Strassen's Matrix Multiplication).

CODECHEF
An unacademy Educational Initiative

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

```
size = int(input())
a = []
b = []
c = []
for i in range(size):
    a.append([int(y) for y in input().split()])
for j in range(size):
    b.append([int(x) for x in input().split()])

m1 = (a[0][0]+a[1][1])*(b[0][0]+b[1][1])
m2 = (a[1][0]+a[1][1])*(b[0][0])
m3 = a[0][0] * (b[0][1]-b[1][1])
m4 = a[1][1] * (b[1][0]-b[0][0])
m5 = b[1][1] * (a[0][0]+a[0][1])
m6 = (a[1][0]-a[0][0]) * (b[0][0]+b[0][1])
m7 = (a[0][1]-a[1][1]) * (b[1][0]+b[1][1])

c0 = m1+m4-m5+m7
c1 = m3+m5
c2 = m2+m4
c3 = m1-m2+m3+m6

c.append([c0,c1])
c.append([c2,c3])

print(c)
```

17:16

Open File ✓Custom Input Run

Custom Input

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

Status Successfully executed **Date** 2021-08-05 04:41:04 **Time** 0.02 sec **Mem** 17.968 kB ×

Input

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

Output

```
[[19, 22], [43, 50]]
```

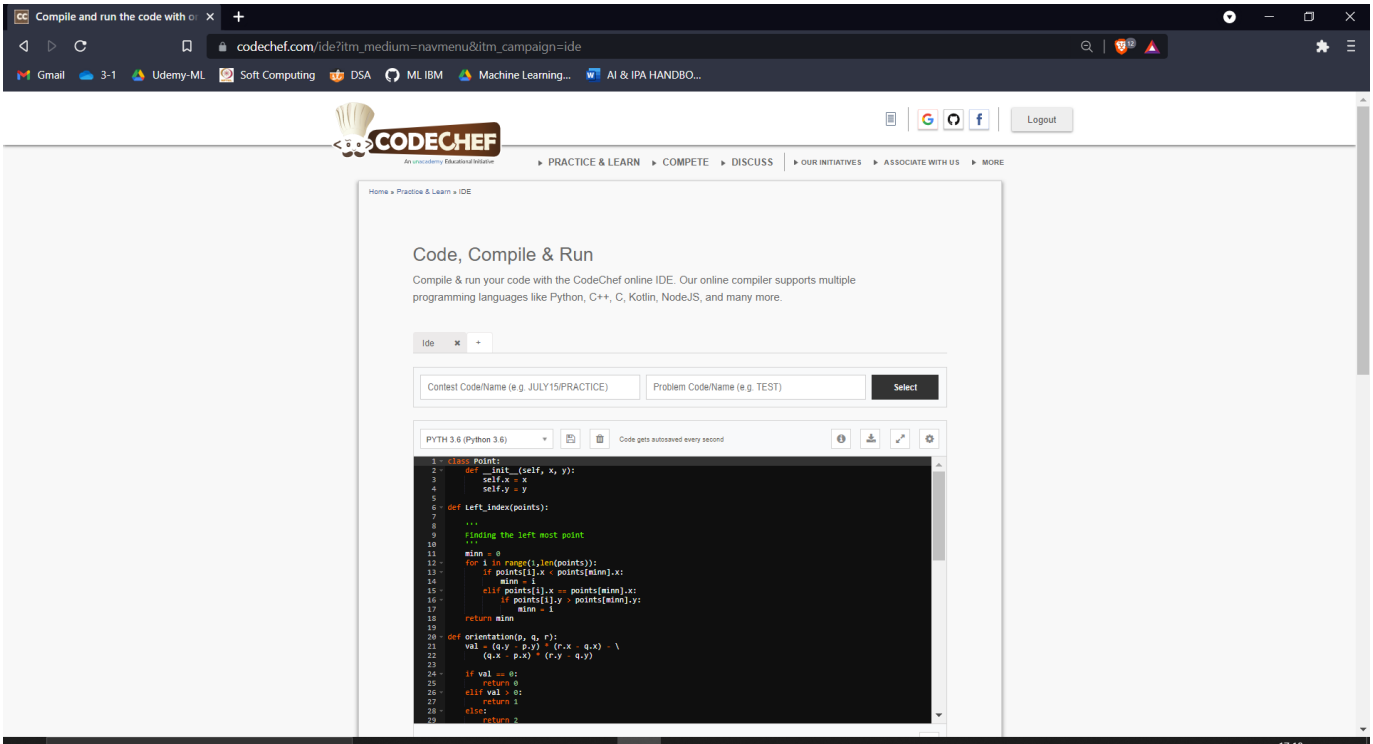
2) Mr. Lee is working in a construction where they had to fencing around trees in a field. The owner has asked a rough estimate to do fencing in that field such all the trees lie inside that region. Consider yourself as a cost estimator who works under Mr. Lee. You are given location of all the trees, and you need to find the points that include this fencing. You need to output the trees that are included in the fencing.

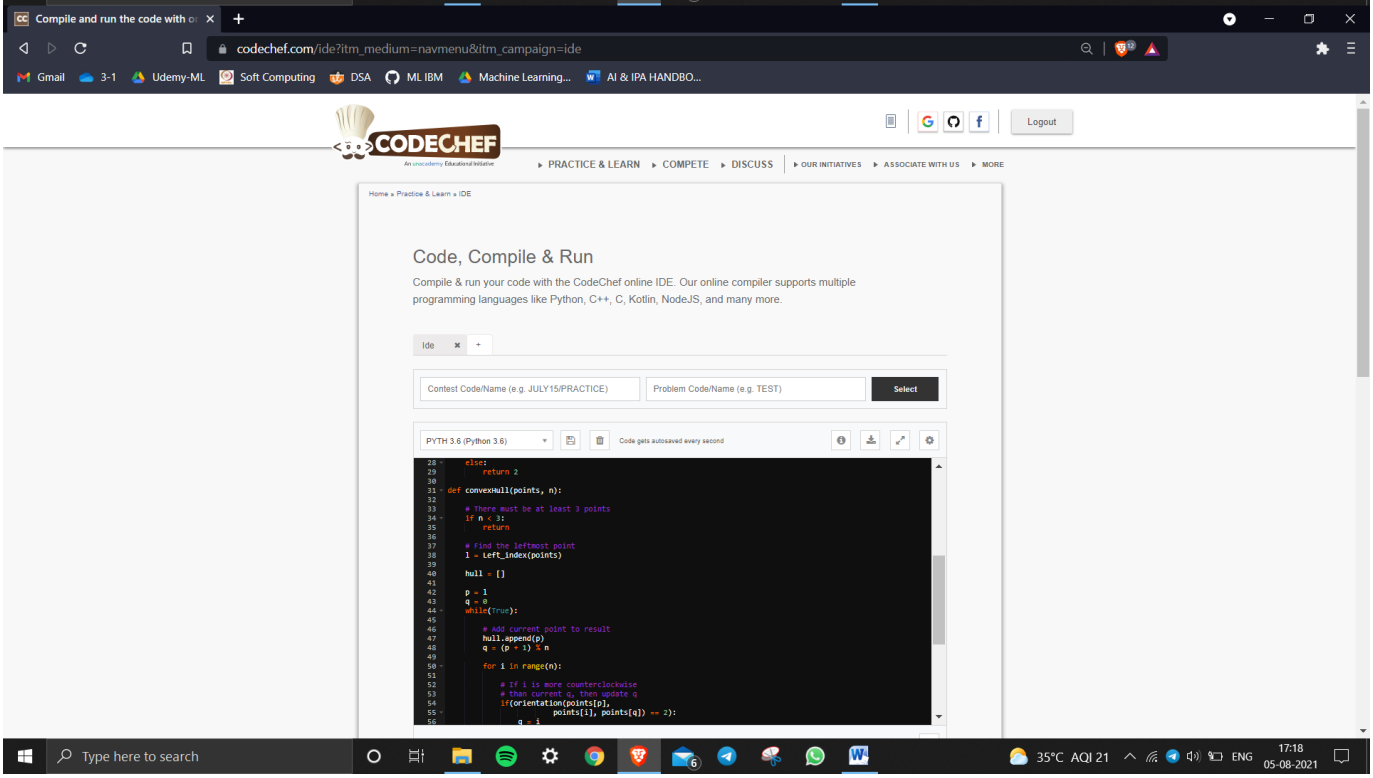
Input:

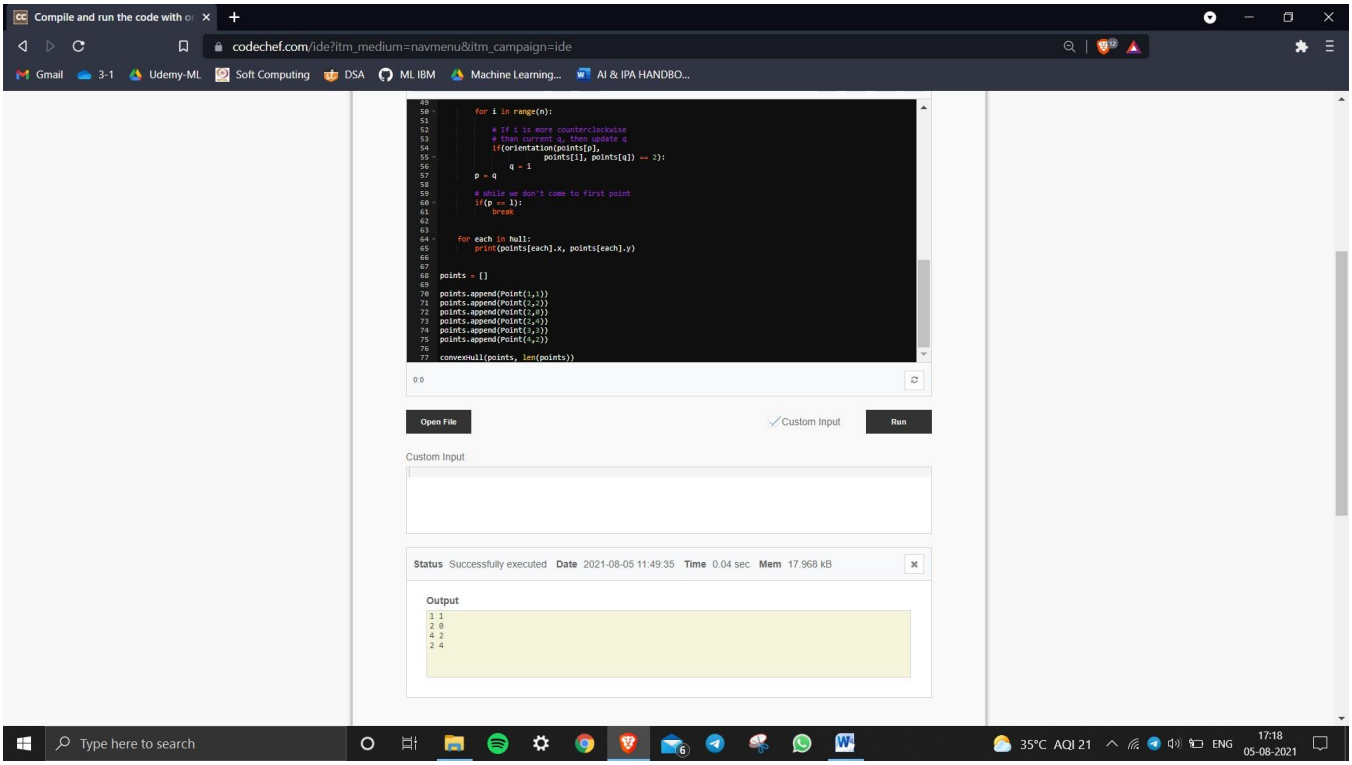
points = [[1,1], [2,2], [2,0], [2,4], [3,3], [4,2]]

Output:

[[1,1], [2,0], [3,3], [2,4], [4,2]]






Compile and run the code with
codechef.com/ide?itm_medium=navmenu&itm_campaign=ide
Gmail
3-1
Udemy-ML
Soft Computing
DSA
ML IBM
Machine Learning...
AI & IPA HANDBO...
for i in range(n):
# If i is more counterclockwise
# than current q, then update q
if(orientation(points[p],
points[i], points[q]) == 2):
q = i
p = q
# While we don't come to first point
if(p == l):
break
for each in hull:
print(points[each].x, points[each].y)
points = []
points.append(Point(1,1))
points.append(Point(2,2))
points.append(Point(2,0))
points.append(Point(2,4))
points.append(Point(3,3))
points.append(Point(4,2))
convexHull(points, len(points))
0:0
Open File
Custom Input
Run
Custom Input
Status Successfully executed Date 2021-08-05 11:49:35 Time 0.04 sec Mem 17.968 kB
Output
1 1
2 0
4 2
2 4
Type here to search
35°C AQI 21
ENG
17:18
05-08-2021

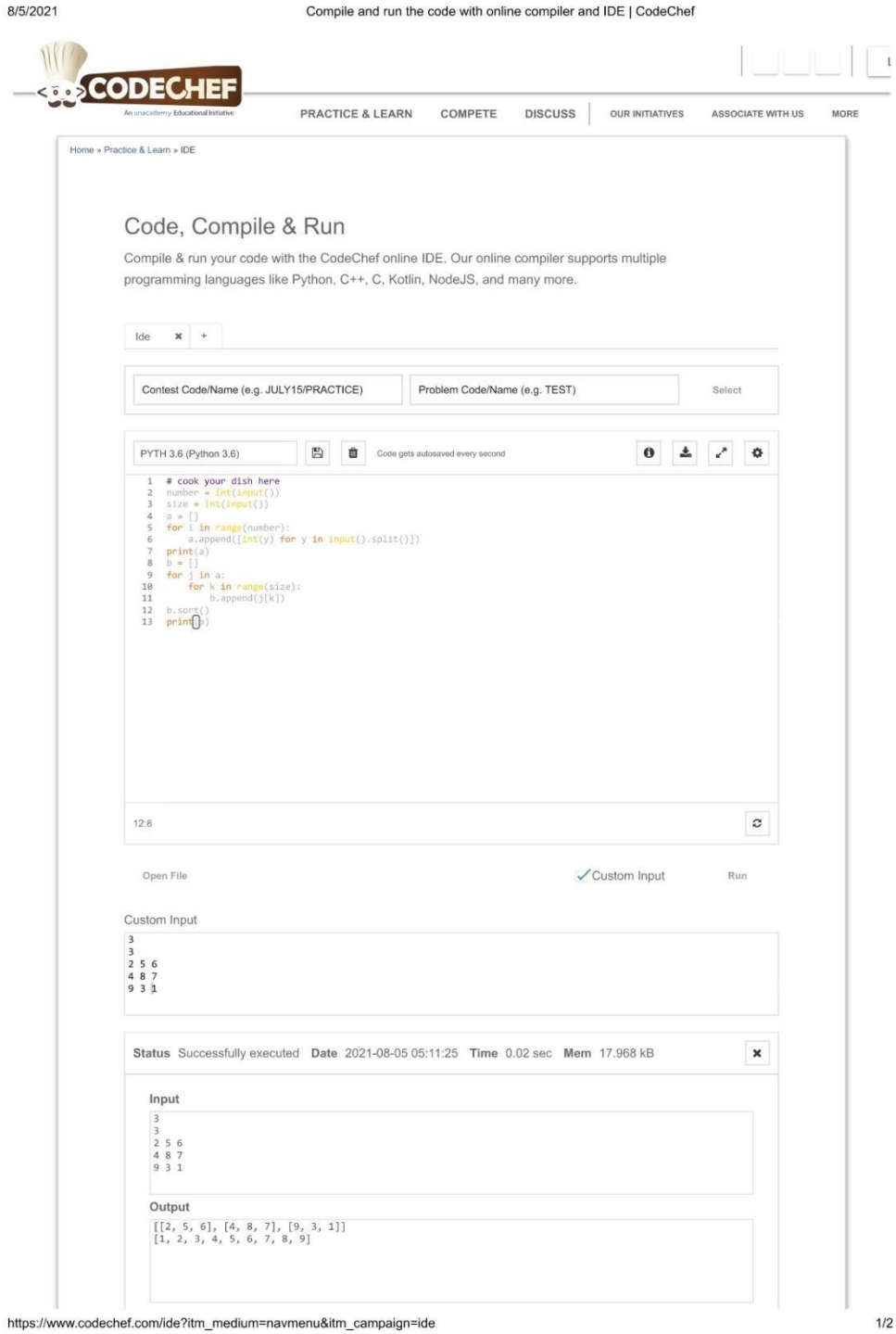
3) In a school there was conducted a contest among two groups. As part of the contest each group must re-arrange the cards that had given to the members in ascending order. Consider yourself as a part of the team and find the best viable way to win that round.

Input

3
3
[[2,5,6], [4,8,7], [9,3,1]]

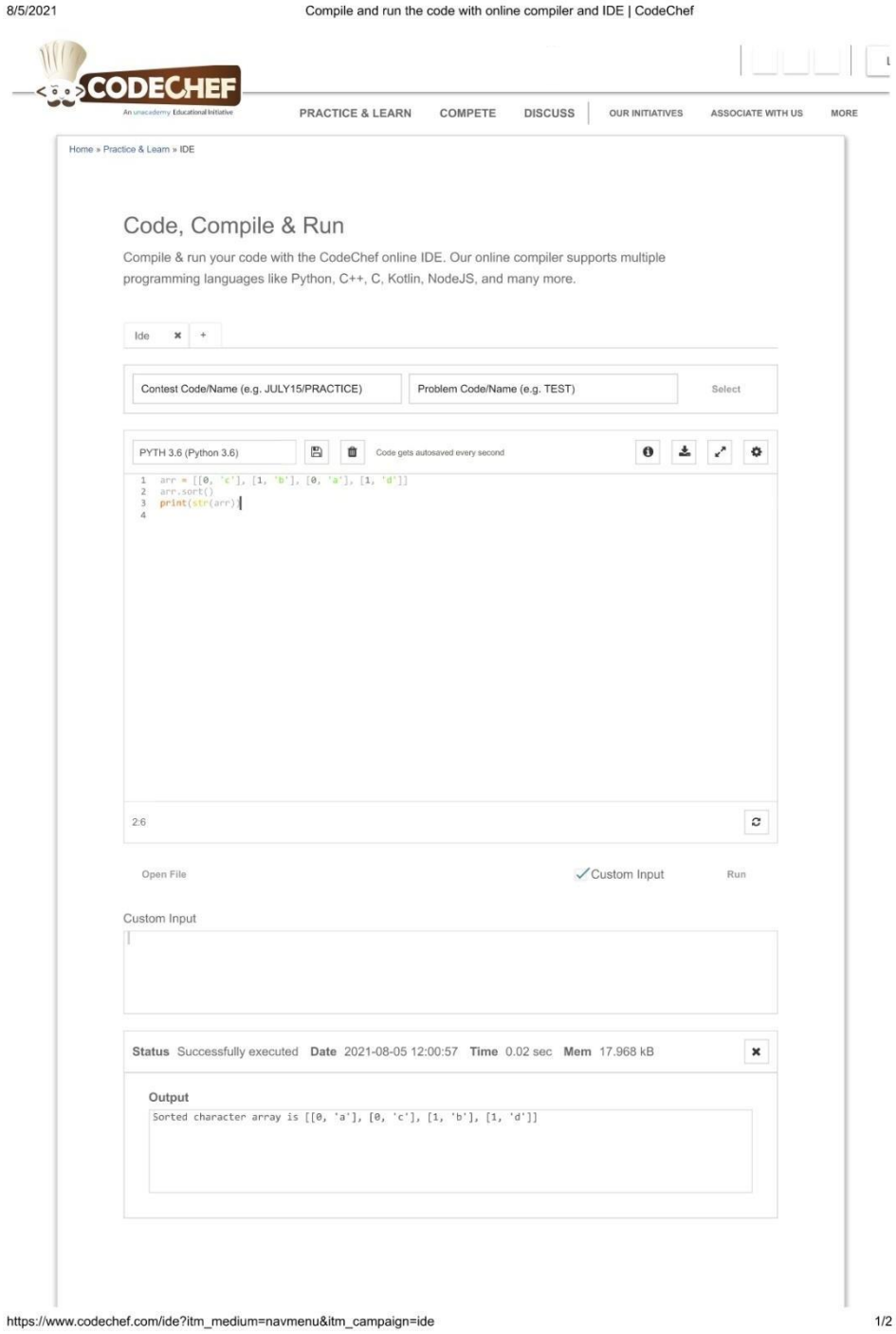
Output

[1,2,3,4,5,6,7,8,9]

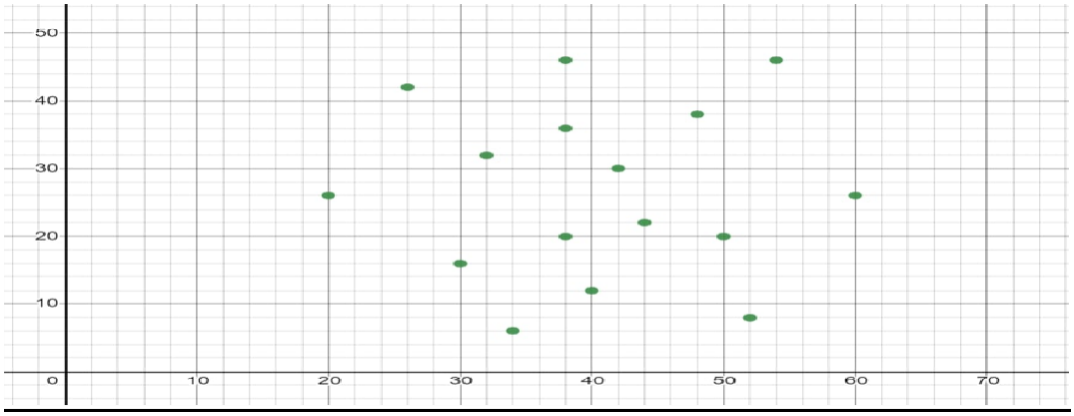


Post-Lab:

1) Mr. Hari Kumar owns a fruit market. In the market there are many sellers who are selling many kinds of fruits. More than one fruits seller can sell same kind of fruit. Mr. Hari Kumar wants to arrange their information in the sorted order based on their names of the sellers and id of the fruits. You must arrange the same type of fruits in the same order as original order.
   0-mangoes 1-apples
   [Hint: Use counting sort algorithm]
   Input
   4
   [[0, c], [1, b], [0, a], [1, d]]
   Output
   [[0, a], [0, c], [1, b], [1, d]]



2) Given a set of points in the plane, apply convex hull algorithm to the given points and explain it step by step process.



2)

Computing a 2-D Convex Hull : Graham's Algorithm

There are many algorithms for computing a 2-D convex hull. The algorithm we will use is Graham's Algorithm which is $O(N \log N)$ algorithm.

Graham's Algorithm is intresting for a no. of reasons:

1. Given N points, find the rightmost, lowestpoint label it P_0.
2. Sort all other points angularly about P_0. Break ties in favour of closeness to P_0. Label the sorted points. $P_1 \cdots P_{N-1}$
3. Push the points labeled P_{N-1} & P_0 onto a stack. These points are guanteed to be on the Convex Hull
4. Set $i=1$
5. while $i<N$ do

   If P_i is strictly left of the line formed by top2 stack entries (P_{top}, P_{top-1}), then push P_i onto the stack & increment i; else pop the stack.

(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: