

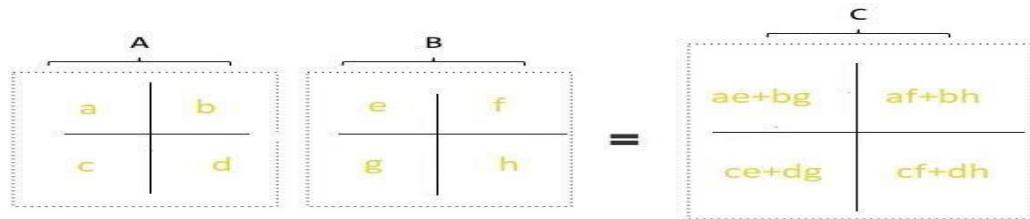
LAB SESSION 04:

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 $N \times N$

a, b, c and d are the sub-Matrices of A of size $N/2 \times N/2$

e, f, g and h are the sub-Matrices of B of size $N/2 \times N/2$

190030319

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

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

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

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

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

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

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

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

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

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

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

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

190030319

Steps:

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

where

$$k = n^2 - 1$$

- 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.

The screenshot shows a web browser window with the CodeChef IDE. The URL is `codechef.com/ide?itm_medium=navmenu&itm_campaign=ide`. The IDE interface includes a top bar with "Compile and run the code with c" and a plus sign. Below the browser window, there's a section for "Contest Code/Name" and "Problem Code/Name" with a "Select" button. The main editor area shows a Java file named "JAVA (HotSpot 8u112)". The code is as follows:

```
1  /* package codechef; // don't place package name! */
2
3  import java.util.*;
4  import java.lang.*;
5  import java.io.*;
6
7  /* Name of the class has to be "Main" only if the class is public. */
8  class Codechef
9  {
10     public static void main (String[] args) throws java.lang.Exception
11     {
12         Scanner sc=new Scanner(System.in);
13         int n=sc.nextInt();
14         int a[]=new int[n];
15         for(int i=0;i<n;i++)
16         {
17             a[i]=sc.nextInt();
18         }
19         int max = a[0];
20         for (int i = 1; i < a.length; i++)
21             if (a[i] > max)
22                 max = a[i];
23
24         int min=a[0];
25         for (int i = 1; i < a.length; i++)
```

At the bottom of the IDE, there are buttons for "Open File", "Custom Input", and "Run". The Windows taskbar at the bottom shows the search bar, task view, and various application icons. The system tray on the right indicates a temperature of 29°C, rain showers, and the date 09-08-2021.

In-Lab:

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

8/5/2021

Compile and run the code with online compiler and IDE | CodeChef

The screenshot displays the CodeChef online IDE interface. At the top, there's a navigation bar with links: PRACTICE & LEARN, COMPETE, DISCUSS, OUR INITIATIVES, ASSOCIATE WITH US, and MORE. Below this, the IDE window is titled "Code, Compile & Run". It shows a Python 3.6 code editor with the following code:

```
1 size = int(input())
2 a = []
3 b = []
4 c = []
5 for i in range(size):
6     a.append([int(y) for y in input().split()])
7     b.append([int(x) for x in input().split()])
8
9
10 m1 = (a[0][0]+a[1][1])*(b[0][0]+b[1][1])
11 m2 = (a[1][0]+a[1][1])*(b[0][0])
12 m3 = a[0][0] * (b[0][1]-b[1][1])
13 m4 = a[1][1] * (b[1][0]-b[0][0])
14 m5 = b[1][1] * (a[0][0]+a[0][1])
15 m6 = (a[1][0]-a[0][0]) * (b[0][0]+b[0][1])
16 m7 = (a[0][1]-a[1][1]) * (b[1][0]+b[1][1])
17
18 c0 = m1+m4-m5+m7
19 c1 = m3+m5
20 c2 = m2+m4
21 c3 = m1-m2+m3+m6
22
23 c.append([c0,c1])
24 c.append([c2,c3])
25
26 print(c)
```

Below the code editor, there's a "Custom Input" section with the following input:

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

The "Status" bar at the bottom indicates "Successfully executed" with a date of "2021-08-05 04:41:04", time of "0.02 sec", and memory of "17.968 kB". The "Input" section shows the input values, and the "Output" section shows the result: `[[19, 22], [43, 50]]`.

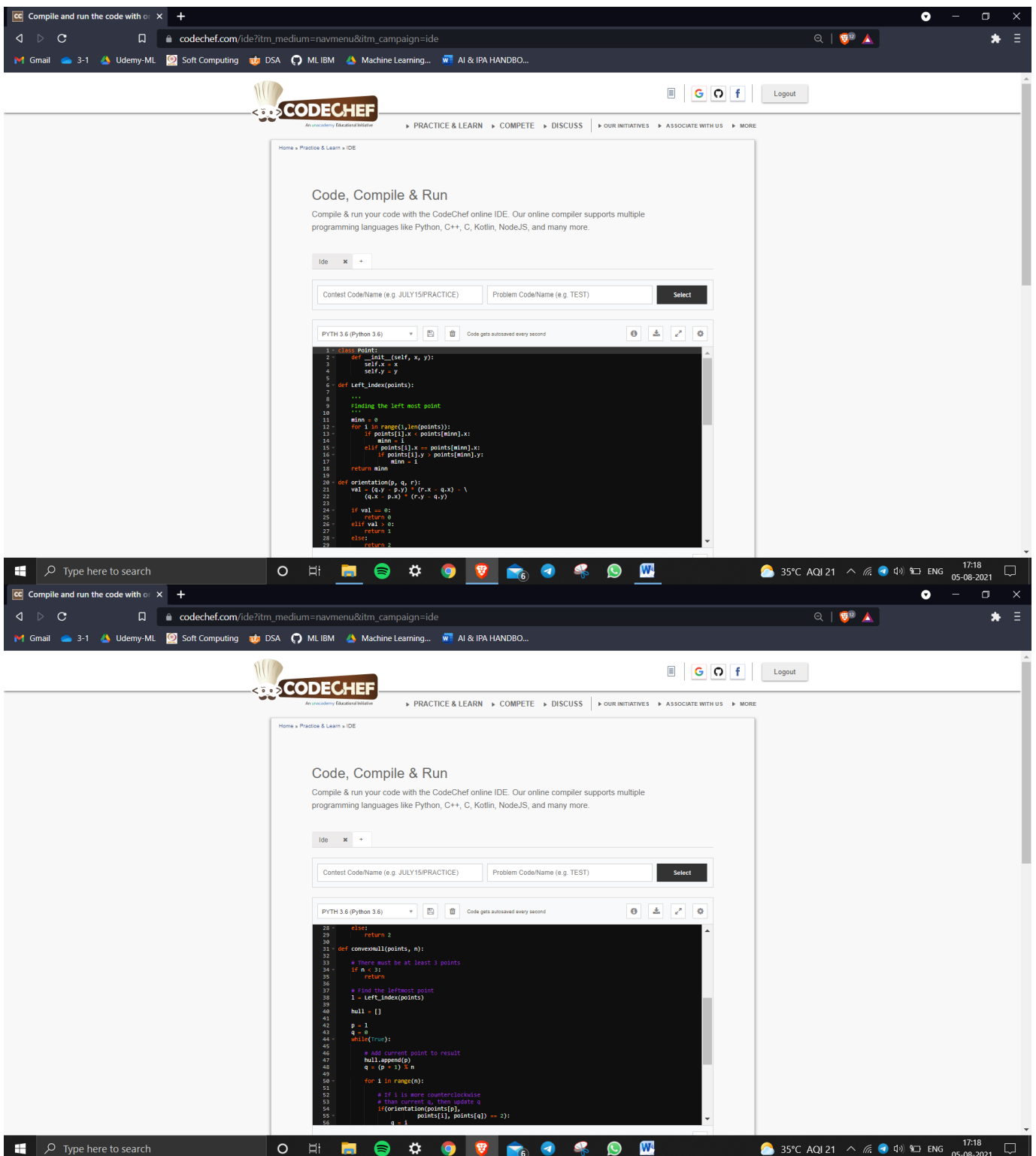
- 2) Mr. Lee is working in a construction where they had to fencing around trees in a field. The owner has asked a rough estimate to do fencing in that field such all the trees lie inside that region. Consider yourself as a cost estimator who works under Mr. Lee. You are given location of all the trees,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

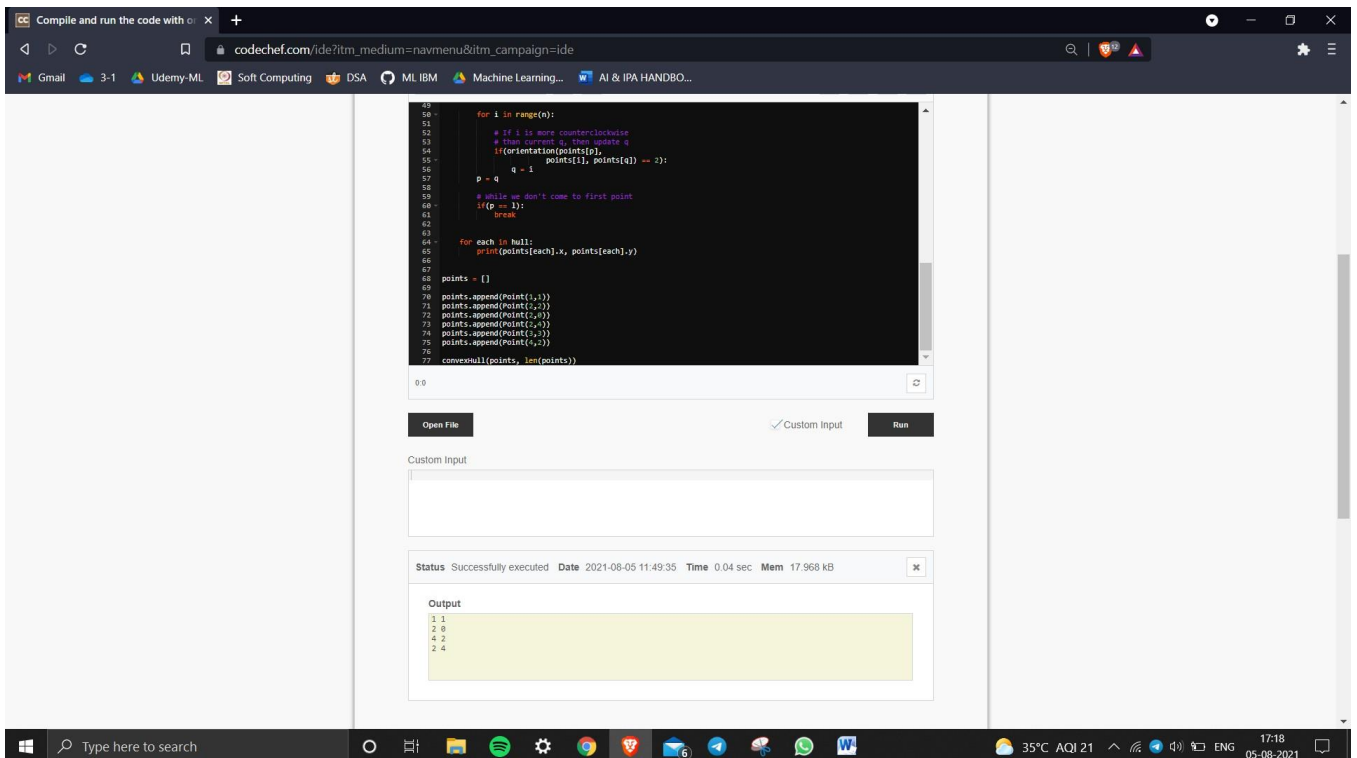
Input:

```
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?tm_medium=navmenu&tm_campaign=ide

Gmail 3-1 Udemyl ML Soft Computing DSA ML IBM Machine Learning... AI & IPA HANDBO...

```
48 for i in range(n):
49     # If i is more counterclockwise
50     # than current q, then update q
51     if(orientation(points[i],
52         points[q], points[i]) >= 2):
53         q = i
54     p = q
55     # While we don't come to first point
56     if(q == 1):
57         break
58     for each in hull:
59         print(points[each].x, points[each].y)
60
61 points = []
62
63 points.append(Point(1,1))
64 points.append(Point(2,2))
65 points.append(Point(2,0))
66 points.append(Point(4,2))
67 points.append(Point(1,3))
68 points.append(Point(4,2))
69
70 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
```

Windows taskbar: Type here to search, 35°C AQI 21, 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

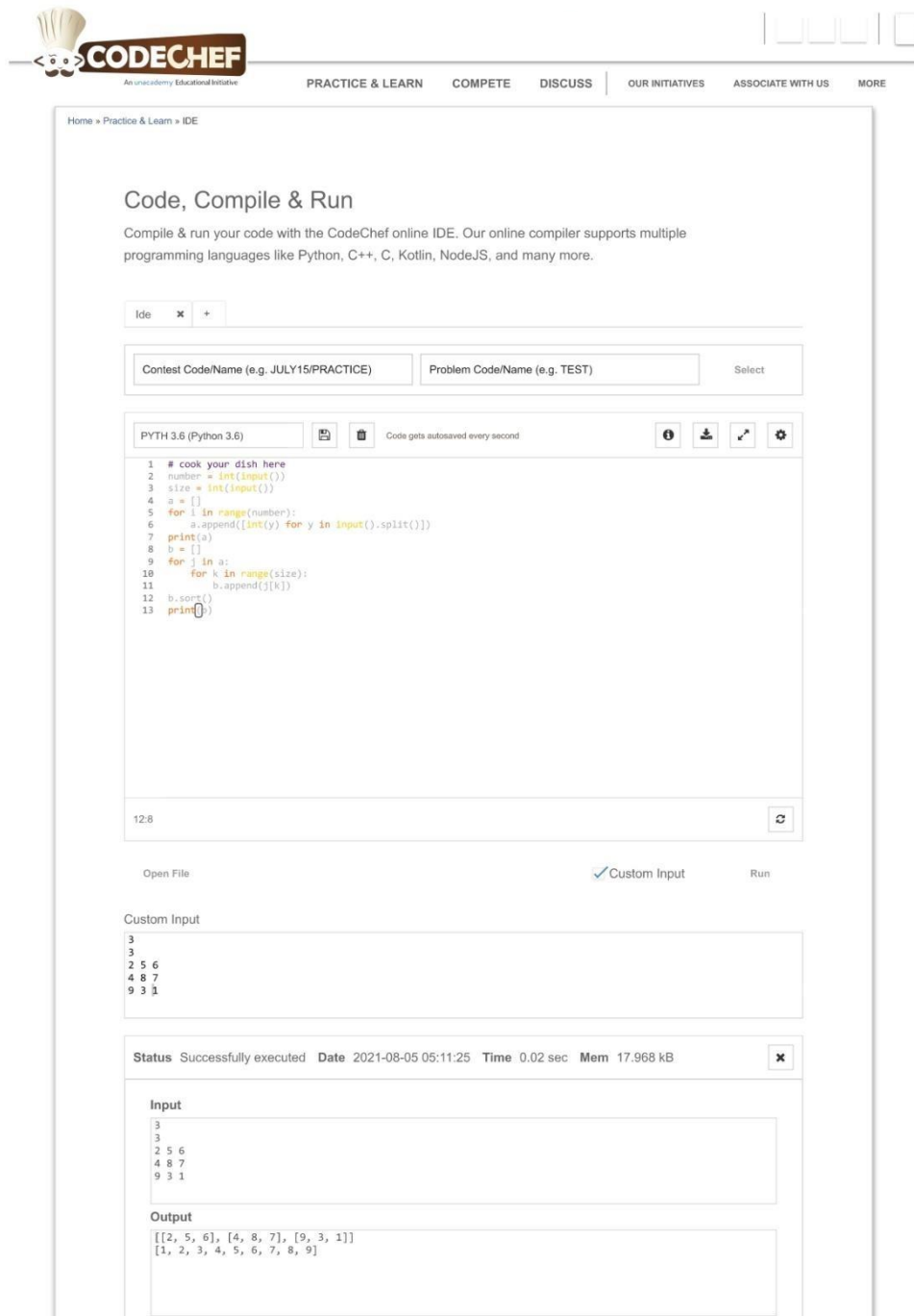
[[2,5,6], [4,8,7], [9,3,1]]

Output

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

8/5/2021

Compile and run the code with online compiler and IDE | CodeChef



The screenshot shows the CodeChef online IDE interface. At the top, there's a navigation bar with links: PRACTICE & LEARN, COMPETE, DISCUSS, OUR INITIATIVES, ASSOCIATE WITH US, and MORE. Below the navigation bar, the main content area is titled "Code, Compile & Run". It contains a text area for code, a "Run" button, and a "Custom Input" checkbox. The code is a Python 3.6 script that reads the number of groups (3), the number of cards per group (3), and the cards themselves. It then sorts the cards in ascending order and prints the result. The output shows the sorted cards: [1, 2, 3, 4, 5, 6, 7, 8, 9].

CodeChef
An e-learning 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

```

1 # cook your dish here
2 number = int(input())
3 size = int(input())
4 a = []
5 for i in range(number):
6     a.append([int(y) for y in input().split()])
7 print(a)
8 b = []
9 for j in a:
10    for k in range(size):
11        b.append(j[k])
12 b.sort()
13 print(b)

```

12:8

Open File ☒ Custom Input Run

Custom Input

```

3
3
2 5 6
4 8 7
9 3 1

```

Status Successfully executed Date 2021-08-05 05:11:25 Time 0.02 sec Mem 17.968 kB

Input

```

3
3
2 5 6
4 8 7
9 3 1

```

Output

```

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

8/5/2021

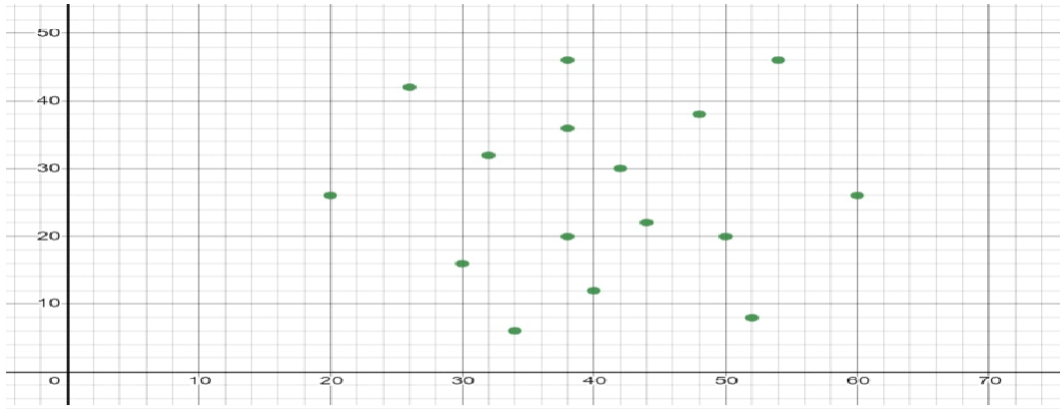
Compile and run the code with online compiler and IDE | CodeChef

The screenshot shows the CodeChef online IDE interface. At the top, there's a navigation bar with links: PRACTICE & LEARN, COMPETE, DISCUSS, OUR INITIATIVES, ASSOCIATE WITH US, and MORE. Below this, the main area is titled "Code, Compile & Run". It contains a code editor with the following Python code:

```
1 arr = [[0, 'c'], [1, 'b'], [0, 'a'], [1, 'd']]
2 arr.sort()
3 print(str(arr))
4
```

Below the code editor, there's a "Custom Input" field with the text: "Sorted character array is [[0, 'a'], [0, 'c'], [1, 'b'], [1, 'd']]". The output section shows the result of the execution: "Sorted character array is [[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 interesting for a no. of reasons:

1. Given N points, find the rightmost, lowest point 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, \dots, P_{N-1} .
3. Push the points labeled P_{N-1} & P_0 onto a stack. These points are guaranteed 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: