icpc international collegiate programming contest

ICPC Europe Regionals 2020

ICPC Northwestern Europe
Regional Contest

Official Problem Set

icpc global sponsor
programming tools

icpc international collegiate programming contest

icpc.foundation

icpc diamond
regional sponsor

Northwestern Europe Regional Contest 2020

NWERC 2020

March 28, 2021

Problems

Do not open before the contest has started.

Problem A
Atomic Energy

The *Next Wave Energy Research Club* is looking at several atoms as potential energy sources, and has asked you to do some computations to see which are the most promising.

Plasma ball by Halacious, Unsplash

Although an atom is composed of various parts, for the purposes of this method only the number of neutrons in the atom is relevant[1]. In the method, a laser charge is fired at the atom, which then releases energy in a process formally called *explodification*. Exactly how this process proceeds depends on the number of neutrons k:

- If the atom contains $k \leq n$ neutrons, it will be converted into a_k joules of energy.
- If the atom contains $k > n$ neutrons, it will decompose into two atoms with i and j neutrons respectively, satisfying $i, j \geq 1$ and $i + j = k$. These two atoms will then themselves explodificate.

When an atom with k neutrons is explodificated, the total energy that is released depends on the exact sequence of decompositions that occurs in the explodification process. Modern physics is not powerful enough to predict exactly how an atom will decompose—however, for explodification to be a reliable energy source, we need to know the minimum amount of energy that it can release upon explodification. You have been tasked with computing this quantity.

Input

The input consists of:

- One line with two integers n and q ($1 \leq n \leq 100$, $1 \leq q \leq 10^5$), the neutron threshold and the number of experiments.
- One line with n integers $a_1, \ldots, a_n$ ($1 \leq a_i \leq 10^9$ for each i), where a_i is the amount of energy released when an atom with i neutrons is explodificated.
- Then q lines follow, each with an integer k ($1 \leq k \leq 10^9$), asking for the minimum energy released when an atom with k neutrons is explodificated.

Output

For each query k, output the minimum energy released when an atom with k neutrons is explodificated.

[1] In fact, for this problem you might want to forget everything you thought you knew about chemistry.

Sample Input 1	Sample Output 1
4 5 2 3 5 7 2 3 5 6 8	3 5 8 10 13

Sample Input 2	Sample Output 2
1 3 10 1 2 100	10 20 1000

Problem B
Bulldozer

You are tasked with bulldozing some buildings that stand along a long, straight road. The buildings are modelled as evenly spaced stacks of identical square blocks along an infinite line. Your powerful bulldozer is capable of moving any one of these blocks one unit of distance to the left or to the right. This may push other blocks out of the way, and blocks which sit atop moving blocks will move along. Blocks which are pushed over a gap fall down until they reach either the ground or another block.

For instance, consider the stacks of blocks shown on the left in Figure B.1 below. If you push the block labelled C to the right, the blocks D and E would be pushed along to the right, since they are in the way. Blocks A, B and F would also move along because they are sitting on top of moving blocks. After pushing C to the right, E would be sitting over a gap, so E and F drop down to fill that gap. The resulting stacks are shown in the middle of Figure B.1. Pushing block C one further step to the right would result in the configuration shown on the right.

Your goal is to *level* all the buildings: bulldoze until all stacks are of height at most 1, i.e., all blocks are on the ground. Note that the road stretches out infinitely far on either side, so this is always possible.

Given the initial heights of the stacks, determine the smallest number of moves you need to make to level all the buildings, where a move consists of using the bulldozer to push one block one step to the left or right.

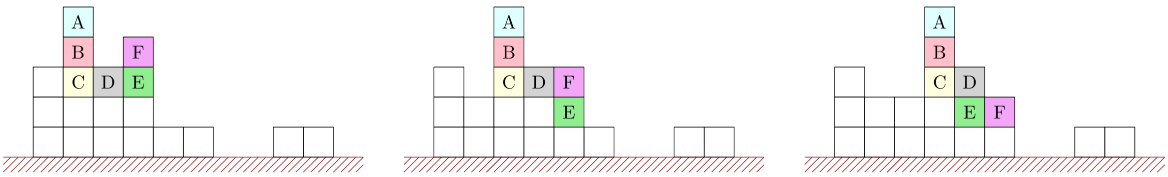


Figure B.1: Illustration of a configuration of stacks of blocks, and the results of pushing the block labelled C towards the right twice (the blocks labelled A–F are coloured and labelled only for illustrative purposes).

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 2 \cdot 10^5$), the number of stacks of blocks.
- One line with n integers $a_1, \ldots, a_n$ ($0 \leq a_i \leq 10^9$ for each i), the initial heights of the stacks from left to right.

The example shown on the left in Figure B.1 could be given by $3, 5, 3, 4, 1, 1, 0, 0, 1, 1$, but it could also be left- or right-padded with additional zeros.

Output

Output the minimum number of moves required to level every building.

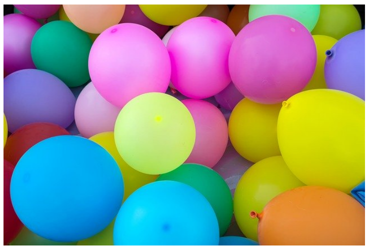
Sample Input 1	Sample Output 1
5 1 1 2 1 1	2

Sample Input 2	Sample Output 2
5 1 4 3 1 1	7

Sample Input 3	Sample Output 3
9 1 0 0 0 6 0 0 0 1	5

Sample Input 4	Sample Output 4
10 1 3 0 0 1 9 1 1 1 1	13

Problem C
Contest Struggles



Balloons by Pexels, Pixabay

Lotte is competing in a programming contest. Her team has already solved k out of the n problems in the problem set, but as the problems become harder, she begins to lose focus and her mind starts to wander.

She recalls hearing the judges talk about the difficulty of the problems, which they rate on an integer scale from 0 to 100, inclusive. In fact, one of the judges said that "*the problem set has never been so tough, the average difficulty of the problems in the problem set is d!*"

She starts thinking about the problems her team has solved so far, and comes up with an estimate s for their average difficulty. In hope of gaining some motivation, Lotte wonders if she can use this information to determine the average difficulty of the remaining problems.

Input

The input consists of:

- One line with two integers n and k ($2 \le n \le 10^6$, $0 < k < n$), the total number of problems and the number of problems Lotte's team has solved so far.
- One line with two integers d and s ($0 \le d, s \le 100$), the average difficulty of all the problems and Lotte's estimate of the average difficulty of the problems her team has solved.

Output

Assuming Lotte's estimate is correct, output the average difficulty of the unsolved problems, or "`impossible`" if the average difficulty does not exist. Your answer should have an absolute or relative error of at most 10^{-6}.

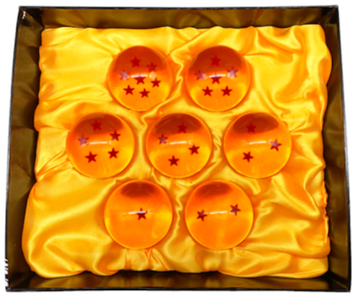
Sample Input 1	Sample Output 1
2 1 70 50	90.00

Sample Input 2	Sample Output 2
10 3 80 90	75.7142857

Sample Input 3	Sample Output 3
2 1 100 10	impossible

This page is intentionally left blank.

Problem D
Dragon Balls



Dragon Balls by Skyringe Crafts

During the showdown with Frieza on Namek, Krillin died yet again and needs to be brought back to life using the *Dragon Balls*. As everybody else is still busy fighting Frieza, the task of retrieving all seven Dragon Balls has fallen to you.

The balls are hidden at unknown locations in a 2D plane and you have been handed a Dragon Radar, designed by Bulma, that you must use to locate them. You can repeatedly fly to arbitrary locations, and the radar will then inform you about the distance to the closest Dragon Ball. If this distance is 0 this means that you found one of the balls and you can then recalibrate the radar so that it ignores the ball you just found.

With the battle still going on, and the radar having limited energy, you are obviously in a great hurry. You need to make sure to collect all the balls by using the radar no more than $1\,000$ times.

Interaction

This is an interactive problem. Your submission will be run against an *interactor*, which reads the standard output of your submission and writes to the standard input of your submission. This interaction needs to follow a specific protocol:

The interactor first sends an integer n ($1 \leq n \leq 7$), the number of Dragon Balls you still need to find. The n balls are hidden at integer locations (x, y) with $0 \leq x, y \leq 10^6$. Your submission may not guess outside this area.

Your submission then repeatedly sends such an integer location (x, y) and the interactor replies with an integer d ($0 \leq d \leq 2 \cdot 10^{12}$), the square of the distance from (x, y) to the closest remaining ball.

If $d = 0$, the ball at (x, y) is considered found and ignored in further guesses. When all balls are found, your submission should exit. Each location holds at most one ball.

Make sure you flush the buffer after each write.

A testing tool is provided to help you develop your solution.

Sample Interaction 1

Read	Write
1	
	0 0
25	
	3 4
2	
	4 3
0	

Read	Sample Interaction 2	Write
`2`		
		`2 1`
`8`		
		`5 5`
`0`		
		`4 2`
`1`		
		`4 3`
`0`		

Problem E
Endgame

The boardgame Chaos is an exotic variant of Chess, played by two players in alternating turns on an $n \times n$ playing board. All pieces have the same set of n valid moves which are agreed on ahead of the game.

Chess by jplenio, CC0 Public Domain

In a single turn a player can pick exactly one of their pieces and perform one of the following actions:

- Perform up to two valid moves using the chosen piece, capturing any piece that the chosen piece lands on along the way.
- Teleport the chosen piece to any cell on the board that is not already occupied by another piece.
- Leave the chosen piece untouched in its current cell.

Having recently discovered Chaos, Alice and Bob are currently in the endgame of a very exciting match. Each player has a single piece left on the board and there are only two turns left, with Alice going next.

Having analysed the situation, she realises that the only way she can win is to capture Bob's piece in her turn. If that is not possible, Alice may be able to force a tie if she can teleport her piece to a cell that Bob cannot capture in his turn. Otherwise Bob will be able to win by capturing Alice's piece, no matter what she does in her turn. Help Alice determine her optimal outcome.

Input

The input consists of:

- One line with an integer n ($2 \leq n \leq 10^5$), the size of the playing board and the number of valid moves.
- One line with two integers a_x and a_y ($1 \leq a_x, a_y \leq n$), the column and row in which Alice's piece is currently located.
- One line with two integers b_x and b_y ($1 \leq b_x, b_y \leq n$), the column and row in which Bob's piece is currently located.
- n lines, the ith of which contains two integers x_i and y_i ($-n < x_i, y_i < n$) representing one of the valid moves. This moves the given piece x_i columns to the right and y_i rows up, provided this does not take the piece outside of the board.

Columns are numbered 1 to n from left to right and rows are numbered 1 to n from bottom to top. All valid moves are distinct.

Output

If Alice can capture Bob's piece in her turn, output "`Alice wins`".

If Alice can use her turn to force a tie by teleporting her piece to a cell that Bob cannot capture in his turn output "`tie`" followed by two integers a'_x and a'_y, the location of any such cell. If there are multiple valid solutions, you may output any one of them.

Otherwise, if Bob is able to capture Alice's piece no matter what she does in her turn, output "`Bob wins`".

Sample Input 1

```
2
2 1
1 2
1 0
0 -1
```

Sample Output 1

```
Bob wins
```

Sample Input 2

```
3
2 3
1 3
-2 1
1 1
1 0
```

Sample Output 2

```
tie 3 1
```

Sample Input 3

```
4
1 1
3 4
0 3
2 0
0 -3
-2 0
```

Sample Output 3

```
Alice wins
```

Problem F
Flight Collision

The *Icelandic Corporation for Parcel Circulation* is the leading carrier for transporting goods between Iceland and the rest of the world. Their newest innovation is a drone link connecting to mainland Europe that has a number of drones travelling back and forth along a single route.

Drone by Hyeri Kim, Pixabay

The drones are equipped with a sophisticated system that allows them to fly evasive manoeuvres whenever two drones come close to each other. Unfortunately, a software glitch has caused this system to break down and now all drones are flying along the route with no way of avoiding collisions between them.

For the purposes of this problem, the drones are considered as points moving along an infinite straight line with constant velocity. Whenever two drones are at the same location, they will collide, causing them to fall off their flight path and plummet into the Atlantic Ocean. The flight schedule of the drones is guaranteed to be such that at no point will there be three or more drones colliding at the same location.

You know the current position of each drone as well as their velocities. Your task is to assess the damage caused by the system failure by finding out which drones will continue flying indefinitely without crashing.

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 10^5$), the number of drones. The drones are numbered from 1 to n.
- n lines, the ith of which contains two integers x_i and v_i ($-10^9 \leq x_i, v_i \leq 10^9$), the current location and the velocity of the ith drone along the infinite straight line.

The drones are given by increasing x coordinate and no two drones are currently in the same position, i.e. $x_i < x_{i+1}$ for each i. You may assume that there will never be a collision involving three or more drones.

Output

Output the number of drones that never crash, followed by the indices of these drones in numerically increasing order.

Sample Input 1	Sample Output 1
3 10 15 30 5 50 -1	1 3

Sample Input 2	Sample Output 2
6 0 3 2 2 3 1 4 3 5 2 6 3	2 1 6

Problem G
Great Expectations

A *speedrun* is a playthrough of a game with the intention to complete it as quickly as possible. When speedrunning, you usually follow a pre-planned path through the game. Along this path, there may be some places where you have to pull off a difficult technique, or *trick*, which may cause a delay if you fail to pull it off successfully. Luckily you can *reset* the game at any time: if you have made a few mistakes, you can start a new run, losing your progress but instantaneously starting over with a clean slate. You can do this as often as you like.

The game you are currently speedrunning has a record of r seconds, which you intend to beat. You have discovered a path through the game that, in the best case, takes $n < r$ seconds. There are some tricks along the way, though: you know exactly where along the run they occur, what the probability is that you will pull them off successfully, and how many seconds you have to spend to recover if they fail.

Given this data, you want to find the optimal strategy for when to reset the game to minimise the expected time to set a new record. Write a program to determine what this smallest possible expected time is.

Input

The input consists of:

- One line with three integers n, r and m ($2 \leq n < r \leq 5\,000$, $1 \leq m \leq 50$), where n and r are as described above and m is the number of tricks.
- m lines, each containing three numbers describing a trick:
  - An integer t ($1 \leq t < n$), the time in the route (assuming no failed tricks before) at which the trick occurs,
  - a real number p ($0 < p < 1$ and p has at most 6 digits after the decimal point), the probability that the trick succeeds, and
  - an integer d ($1 \leq d \leq 1\,000$), the number of seconds required to recover in case the trick fails.

The tricks are given in sorted order by t, and no two tricks occur at the same time t in the route.

You may assume that, without resetting, a single playthrough has a probability of at least 1 in $50\,000$ to succeed at improving the record.

Output

Output the expected time you will have to play the game to set a new record, assuming an optimal strategy is used. Your answer should have an absolute or relative error of at most 10^{-6}.

Explanation of Sample Input 1

The record for this game is 111 seconds, and your route takes 100 seconds if everything goes right.

After playing for 20 seconds, there is a trick with a 50% success rate. If it succeeds, you keep playing. If it fails, you incur a 10 second time loss: now the run will take at least 110 seconds. It is still possible to set a record, but every other trick in the run has to be successful. It turns out to be faster on average to reset after failing the first trick.

Thus you repeat the first 20 seconds of the game until the trick is successful: with probability $1/2$, it takes 1 attempt; with probability $1/4$, it takes 2 attempts; and so on. On average, you spend 40 seconds on the first 20 seconds of the route.

Once you have successfully performed the first trick, you want to finish the run no matter the result of the other tricks: it takes 80 seconds, plus on average 1 second loss from each of the remaining 4 tricks. So the expected time until you set a record is 124 seconds.

Sample Input 1

```
100 111 5
20 0.5 10
80 0.5 2
85 0.5 2
90 0.5 2
95 0.5 2
```

Sample Output 1

```
124
```

Sample Input 2

```
2 4 1
1 0.5 5
```

Sample Output 2

```
3
```

Sample Input 3

```
10 20 3
5 0.3 8
6 0.8 3
8 0.9 3
```

Sample Output 3

```
18.9029850746
```

Sample Input 4

```
10 50 1
5 0.5 30
```

Sample Output 4

```
15
```

Problem H
Hot Springs

Blue Lagoon by Andy Wang, Unsplash

Iceland is famous for its geothermal activity, which supplies the country with much of its electricity and heating. It is also a source of pleasure, in the form of hot springs.

Kalle is visiting one of Iceland's famous hot springs. It contains n pools of water, where the ith one has temperature t_i. Although staying in one of the warmer pools for a long time sure is relaxing, Kalle is on a very tight schedule and just wants a quick dip in each of the pools. As you may know, the nicest thing about hot baths is the contrast between hot and cold. Therefore, to get the most out of his stay, Kalle wants to find an ordering of the pools so that the difference in temperature between subsequent pools is increasing.

Given a sequence of pool temperatures $t_1, t_2, \ldots, t_n$, rearrange them into a new sequence $t'_1, t'_2, \ldots, t'_n$ such that for all $2 \leq i \leq n-1$ it holds that

$$|t'_{i-1} - t'_i| \leq |t'_i - t'_{i+1}|.$$

Input

The input consists of:

- One line with an integer n ($2 \leq n \leq 10^5$), the number of pools.
- One line with n integers $t_1, \ldots, t_n$ ($-10^5 \leq t_i \leq 10^5$ for each i), the temperatures in each of the n pools.

Output

Output a rearrangement of the sequence satisfying the given requirement. If no solution exists, output "`impossible`". If there are multiple valid solutions, you may output any one of them.

Sample Input 1	Sample Output 1
3 1 3 4	4 3 1

Sample Input 2	Sample Output 2
6 0 0 1 -1 -6 3	0 1 3 -1 -6 0

This page is intentionally left blank.

Problem I
Island Tour

Iceland by Irena Jackson, CC0 Public Domain

Tijmen, Annemarie and Imme are visiting Iceland, a beautiful island country located in the middle of the Atlantic Ocean. To see as much of the island as possible, they would like to visit all of the tourist attractions on the Ring Road; the main road that runs around the circular perimeter of the island. There are n attractions, conveniently numbered from 1 to n in the order they appear along the road.

Unfortunately, current distancing measures only allow one visitor at a time at any given attraction, so they have decided to split up. Each person will start at a different attraction, visiting the remaining attractions in circular order around the Ring Road, i.e. a person starting their tour at attraction i visits the attractions in the order i, $i+1$, …, n, 1, …, $i-1$.

They know how long it takes to travel from one attraction to the next and how much time each person is going to spend at each attraction. They will each start their tour at the same time and—due to their impatience—will follow their plan without any waiting. Help Tijmen, Annemarie and Imme decide where each person should start their tour such that there never comes a time where more than one person is located at the same attraction. A person may enter an attraction at the same moment another person leaves the attraction, and when a person is finished visiting their last attraction they will immediately leave the attraction and return to their hotel.

Input

The input consists of:

- One line with an integer n ($1 \leq n \leq 400$), the number of tourist attractions.
- One line with n integers $d_1, \ldots, d_n$ ($1 \leq d_i \leq 10^6$ for each i), where d_i is the travel time in minutes from tourist attraction i to $i+1$ (or to 1 when $i = n$).
- For each of Tijmen, Annemarie and Imme:
  - One line with n integers $t_1, \ldots, t_n$ ($1 \leq t_i \leq 10^6$ for each i), where t_i is the time in minutes that the given person is going to spend at attraction i.

Output

If there is a valid assignment, output one line with three integers, the starting attraction for each person. Otherwise, output "`impossible`". If there are multiple valid solutions, you may output any one of them.

Sample Input 1	Sample Output 1
`` 6 1 1 1 1 1 1 2 1 3 2 3 1 8 7 4 9 7 2 7 6 2 9 2 1 ```	``` 1 5 6 ```

Sample Input 1

```
6
1 1 1 1 1 1
2 1 3 2 3 1
8 7 4 9 7 2
7 6 2 9 2 1
```

Sample Output 1

```
1 5 6
```

Sample Input 2

```
4
1 1 1 1
1 1 1 1
10 3 2 1
4 2 5 1
```

Sample Output 2

```
impossible
```

Problem J
Joint Excavation

Mole by Ahmad Kanbar, Unsplash

The mole family recently decided to dig a new tunnel network. The layout, which has already been decided, consists of chambers and bidirectional tunnels connecting them, forming a connected graph. Mother mole wants to use the opportunity to teach her two mole kids how to dig a tunnel network.

As an initial quick demonstration, mother mole is going to start by digging out a few of the chambers and tunnels, in the form of a non-self-intersecting path in the planned tunnel network. She will then divide the remaining chambers between the two mole kids, making sure that each mole kid has to dig out the same number of chambers, or else one of the mole kids will become sad. (The tunnels are much easier to dig out, and thus of no concern.) The kids may work on their assigned chambers in any order they like.

Since the mole kids do not have much experience with digging tunnel networks, mother mole realises one issue with her plan: if there is a tunnel between a pair of chambers that are assigned to different mole kids, there is a risk of an accident during the excavation of that tunnel if the other mole kid happens to be digging in the connecting chamber at the same time.

Help mother mole decide which path to use for her initial demonstration, and how to divide the remaining chambers evenly, so that no tunnel connects a pair of chambers assigned to different mole kids. The initial path must consist of at least one chamber and must not visit a chamber more than once.

Input

The input consists of:

- One line with two integers c and t ($1 \leq c \leq 2 \cdot 10^5$, $0 \leq t \leq 2 \cdot 10^5$), the number of chambers and tunnels in the planned tunnel network.
- t lines, each containing two integers a and b ($1 \leq a, b \leq c$, $a \neq b$), describing a bidirectional tunnel between chambers a and b.

The chambers are numbered from 1 to c. There is at most one tunnel between any pair of chambers, and there exists a path in the network between any pair of chambers.

Output

First output two integers p and s, the number of chambers on the path in mother mole's initial demonstration and the number of chambers each mole kid has to dig out. Then output a line containing the p chambers in mother mole's initial path, in the order that she digs them out. Then output two more lines, each containing the s chambers that the respective mole kid has to dig out, in any order.

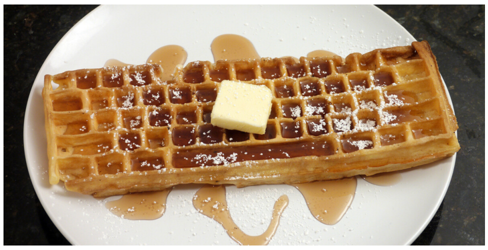
The input is chosen such that there exists at least one valid solution. If there are multiple valid solutions, you may output any one of them.

Sample Input 1	Sample Output 1
``3 2``` ```3 1``` ```2 1```	```3 0``` ```3 1 2```

Sample Input 2	Sample Output 2
``4 3``` ```1 3``` ```2 3``` ```3 4```	```2 1``` ```3 4``` ```2``` ```1```

Sample Input 3	Sample Output 3
``7 7``` ```1 2``` ```2 3``` ```4 2``` ```2 5``` ```4 5``` ```6 7``` ```7 2```	```3 2``` ```5 2 7``` ```6 1``` ```3 4```

Problem K
Keyboardd



Waffle keyboard by The Keyboard Waffle Iron

Oh no! Joanna just spilled some syrup on her keyboard and now some of the keys are sticky. This causes her considerable frustration, as every time she presses one of the sticky keys, the corresponding character gets entered twice on her computer.

This could not have happened at a more inconvenient time; it is the start of the contest and she was just about to type in the solution to the first problem! Joanna does not have time to remove and clean every key on her keyboard, so she wonders if there is a way to quickly identify the sticky keys. Starting to panic, she quickly types some text on her keyboard and stares at the resulting text displayed on her screen.

Given the line of text that Joanna typed on her keyboard and the resulting text displayed on her screen, help her determine which keys must be sticky.

Input

The input consists of:

- One line containing a string s ($1 \leq \text{length}(s) \leq 1\,000$), the text that Joanna typed on her keyboard.
- One line containing a string t ($1 \leq \text{length}(t) \leq 1\,000$), the text displayed on Joanna's screen as a result.

Both s and t consist only of lower-case letters (‘`a`’–‘`z`’) and spaces (‘ ’), and start and end with a letter.

It is guaranteed that t is the result of doubling each character in s that corresponds to a sticky key. At least one character in s corresponds to a sticky key (i.e. $s \neq t$).

Output

Output all characters (letters and space) corresponding to keys that must be sticky, in any order.

Sample Input 1	Sample Output 1
`this is very annoying` `thiss iss veery annoying`	`se`

Sample Input 2	Sample Output 2
`so sticky` `ssoo  ssttiicckkyy`	`its yock`

This page is intentionally left blank.