Contest	[Fall17][B1] Assignment #1
Problem	[Integration] Point in Polygon
Input file(s)	`input.txt`
Output file	`output.txt`
Time limit	20 sec.
Memory limit	128 MB

Statement

1. Implement your own `Crossing number algorithm` class, for Point in Polygon .

2. Use Monte Carlo method of integration to calculate the area of the polygon. (This require previously implemented method "point in polygon").
Set integration accuracy to 1e-4: if next iteration of Monte-Carlo simulation change area **by less than 1e-4** - you must finish the loop.

`Hint:` For point of interest, we build a random ray. If this ray intersects polygon even number of times (0, 2, 4, ...) – point is outside the polygon, else point is inside. Your implementation should be something like static Boolean inside (YourListType polygon, Point2D point) { ... }.

Inside `input.txt` there are `double` coordinates of polygon like `(x1 y1) (x2 y2) (x3 y3)...`.
Write your answer to `output.txt`. Your answer should contain Yes|No (if point inside - Yes) and Area as in example below

Example

input.txt

```
{(0,0),(10,0),(10,10),(0,10)}
(20,20)
```

`output.txt`

```
No

100
```

Contest	[Fall17][B1] Assignment #1
Problem	[QUEUE] Josephus Prob.
Input file(s)	`input.txt`
Output file	`output.txt`
Time limit	10 sec.
Memory limit	128 MB

Statement

1. Implement `LinkedQueue` class.

2. Using your Queue Implementation solve Josephus Problem stated below.

There are n people standing in a circle waiting to be executed. After k - 1 people are skipped and the k^th man is executed. Then again, k-1 people are skipped and the k^th man is executed. The elimination proceeds around the circle (which is becoming smaller and smaller as the executed people are removed), until only the last man remains, who is given freedom. The task is to choose the place in the initial circle so that you survive (remain the last one), given n and k.

Inside `input.txt` there are some names (<= 10) and k (amount of people cycled through before deleted. Write the last name to `output.txt`.

Example

`input.txt`

```
Name1 Name2 Name3 4
```

---

output.txt

```
Name2
```

Contest	[Fall17][B1] Assignment #1
Problem	[STACK] Shunting-Yard
Input file(s)	`input.txt`
Output file	`output.txt`
Time limit	10 sec.
Memory limit	128 MB

Statement

1. Implement `LinkedStack` classes.

2. Implement Shunting-yard algorithm for Arithmetic operations (“+ - * /”) and brackets (“()”), No functions handling required.

3. Implement an evaluation of Reverse Polish Notation (RPN) for double values.

If there are error in input – program should print “ERROR” to the output file.

Read the expression from the `input.txt` file. Write results to `output.txt` rounded to 2 decimal digits after the point.

Example

---

input.txt

```
(1 + 5)/(7 + 1)
```

---

output.txt

```
0.75
```