Testing

inputted values	Expected output	Actual meet
-14 -99	wrong input	Yes
3 9999	wrong input	Yes
a.out -r3 -test	print usage info	Yes
a.out -r 5 -m 3 -c 4	Print an grid with row:5; mine3; cols:4	Yes

Design

1. 2D Dynamic Array
   '0 - 8' hint
   'x' flag
   '*' boom!
   random
   hidden

```
0 0 0
0 X 0
0 0 0
[ ] [ ] [ ]
[ ] [ ] [ ]   √
[ ] [ ] [ ]
```

2. Play again
   1. Yes -> play again <- <struct>
   2. No -> exit

3. error
   - invalid input checker
     - input a cell which has already open.
       return to choose
     - invalid row / colum for flag / open
       return to choose.

* random mine

mine => '*'

random put '*' to

[ ] [ ] [ ]
[ ] [ ] [ ]
[ ] [ ] [ ]

(1) random number generator: 0 or 1 ✓

<2> if else statement ✓

if =0 => '0' || else => '*'

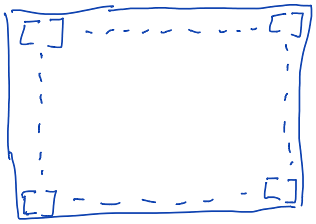
<3> put [ ] and [*] =>

[ ] [ ] [ ]
[ ] [*] [*]
[ ] [ ] [ ]
- - - - -

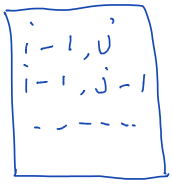
if { ..... } else :

Scan : [0] -> if mine exist around , 0++ (*mine)

Special situations :



if open '0' : check around 8.



if is '0' , open.

a.out -r 9 -c 9 -m 10

a.out

How many row, colums? 9 □ 9 ↵

mines? 10 ↵

open(1)orflag(2)?

row, col ?->9 0

Cin >> rows >> cols ;

#rows

#columns

#mines

/a.out -r 3 -c 5 -m 7

* if sth after dash != r, c, m

* dashes have to before int.

* # argc is correct.