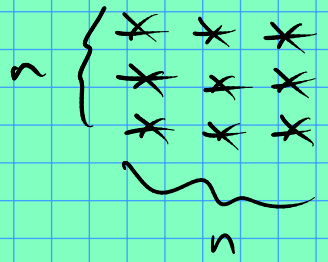
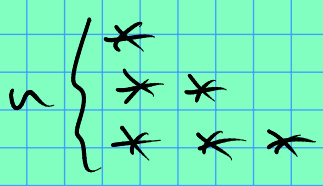
Exercise: on input n (int),
print this pattern:




```
int main()
{
    int n;
    cin >> n;
    for (int i=0; i<n; i++) {
        // print a row
        for (int j=0; j<n; j++) {
            cout << "* ";
        }
        cout << endl;
    }
}
```

```
cout << " n == " << n << "\n";
printf("i == %i\n", n);
```

Exercise 2: input n. output:



```
int main()
{
    int n;
    cin >> n;
    for (int i=0; i<n; i++) {
        // print i+1 *'s
        for (int j=0; j<=i; j++) {
            cout << "*";
        }
        cout << endl;
    }
}
```

Ex: print first even number from stdin:

```
int n;
while(cin>>n) {
    if(n%2==0) {
        cout << n;
        break;
    }
}
```

Exercise: find all right triangles w/

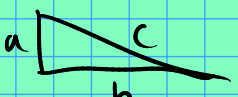
① integer sides

② perimeter $\leq p$ ($\leftarrow$ only input!)

Idea: use "brute force".

How to enumerate possible list of solutions?

Option 1: check all $1 \le a \le b \le c \le p/2$



Option 2: check $1 \le a \le b \le p/2$, but also check $\sqrt{a^2+b^2}$ for being an integer...