In [11]:

```python
# Hallow Square:

n = int(input())
for r in range(1,n+1):
    for c in range(1,n+1):
        if r == 1 or c == 1 or r==n or c==n:
            print("{}".format("*"),end=" ")
        else:
            print("{}".format("&"),end=" ")
    print(end="\n")
```

```
6
* * * * * *
* & & & & *
* & & & & *
* & & & & *
* & & & & *
* * * * * *
```

In [22]:

```python
# Diagonals

n = int(input())
for i in range(1,n+1):
    for j in range(1,n+1):
        if i+j==n+1 or i==j:
            print("{}".format("*"),end=" ")
        else:
            print("{}".format("@"),end=" ")
    print(end="\n")
```

```
5
* @ @ @ *
@ * @ * @
@ @ * @ @
@ * @ * @
* @ @ @ *
```

Jumping statements:

- To execute the flow of execution based on condition it has to skip or to ter minate from the loop.
- It should be used within a loop[conditional statments]
-> continue -> It skips the value and executes untill loop ends
-> break -> It breaks from the condition when it is true and it doesnt repeats untill end
-> pass -> It skips the line[comment] -> function
-> return -> It returns the value from function to main

In [24]:

```
n = int(input())
for r in range(n+1):
    if r%2==0:
        print("{}".format(r),end=" ")
print(r)
```

```
5
0 2 4 5
```

In [31]:

```
n = int(input())
for k in range(n+1):
    if k%2!=0:
        print("{}".format(k),end=" ")
        break
print(k)
```

```
9
1 1
```

In [37]:

```
m = int(input())
for l in range(m+1):
    if l == 0:
        continue
    else:
        print("{}".format(l),end=" ")
        break
print(l)
```

```
100
1 1
```

In [ ]:

```

```