

# Triangular Pattern

Pattern 1 :

```
1
12
123
1234
```

- 1) n=4
- 2) All ith row have ith columns.
- 3) What to print is depend on jth column.Means for first column print 1, so on...

In [10]:

```
n = int(input())
i = 1
while(i <= n):
    j = 1
    # j depend on ith value
    while(j<=i):
        print(j,end='')
        j = j + 1
    print()
    i = i + 1
```

```
4
1
12
123
1234
```

In [17]:

```
# via temporary variable k
n = int(input())
i = 1
while(i <= n):
    k = 1
    j = 1
    # j depend on ith value
    while(j<=i):
        print(k,end='')
        k = k + 1
        j = j + 1
    print()
    i = i + 1
```

```
4
1
12
123
1234
```

Pattern 2 :

```
1
23
345
4567
```

1) n=4

2) All ith row have ith columns.

3) What to print is depend on temporary value k which start with intial value of ith row, and print until loop running and print value equal to row number times. E.g. For first row 1 elements, for 2nd row 2 elements , so on..

.We do need to reset the value of k since it depedent from row number.

In [11]:

```
# by temp variable
n = int(input())
i = 1
while(i <= n):
    j = 1
    k = i
    while(j<=i):
        print(k,end='')
        j = j + 1
        k = k + 1
    print()
    i = i + 1
```

```
4
1
23
345
4567
```

In [12]:

```
# by formula
n = int(input())
i = 1
while(i <= n):
    j = 1
    while(j<=i):
        print(i+j-1,end='')
        j = j + 1
    print()
    i = i + 1
```

```
4
1
23
345
4567
```

Pattern 3 :

```
1
23
456
78910
```

- 1) n=4
- 2) All ith row have ith columns.
- 3) What to print is depend on temporary variable k which held the current number and keep incrementing it. We don't need to reset the value of k since it independent from row number.

In [16]:

*#We don't need to reset the value of k since it independent from row numbe*

```

n = int(input())
i = 1
while(i <= n):
    k = 1
    j = 1
    while(j<=i):
        print(k,end='')
        k = k + 1
        j = j+1
    print()
    i = i + 1

```

```

4
1
12
123
1234

```

In [13]:

```

n = int(input())
i,k = 1,1
while(i <= n):
    j = 1
    while(j<=i):
        print(k,end='')
        k = k + 1
        j = j+1
    print()
    i = i + 1

```

```

4
1
23
456
78910

```

In [14]:

```
# spaced version
n = int(input())
i,k = 1,1
while(i <= n):
    j = 1
    while(j<=i):
        print(k,end=' ')
        k = k + 1
        j = j+1
    print()
    i = i + 1
```

```
4
1
2 3
4 5 6
7 8 9 10
```

## Character Patterns

### print kth alphabets

**kth alphabet = 'A' + k - 1**

For that we use ord() and chr() methods.

1) ord() = Give ascii code of character. It's only works with one character,

2) chr() = Give character according to ascii code.

In [8]:

```
ord('A')
```

Out[8]:

65

In [9]:

```
ord('a')
```

Out[9]:

97

In [10]:

```
ord('as')
```

-----  
-----

**TypeError**

Traceback

```
(most recent call last)  
<ipython-input-10-b457ae31090b> in <module>  
----> 1 ord('as')
```

**TypeError:** ord() expected a character, but string of length 2 found

In [11]:

```
chr(32)
```

Out[11]:

' '

In [12]:

```
chr(97)
```

Out[12]:

'a'

In [13]:

```
#print kth alphabet
k = int(input())
x = ord('A')
ascii_hold = x + k - 1
char_target = chr(ascii_hold)
char_target
```

5

Out[13]:

'E'

In [14]:

```
# in one line
k = int(input())
char_target = chr(ord('A') + k - 1)
char_target
```

5

Out[14]:

'E'

Pattern 1:

ABCD  
ABCD  
ABCD  
ABCD

1) n = 4  
2 ) j =4  
3) print 'A'+j - 1

In [23]:

```

num = int(input())
i = 1
while(i <= num):
    j = 1
    while(j <= num):
        char_target = chr(ord('A') + j - 1)
        print(char_target, end='')
        j = j + 1
    print()
    i = i + 1

```

```

4
ABCD
ABCD
ABCD
ABCD

```

Pattern 2:

```

ABCD
BCDE
CDEF
DEFG

```

1) n = 4  
 2 ) j = 4  
 3) print 'A'+j - 1 . Here, not every row start with A show we need a one start\_char for every row. Show for first row it is A, the n B for second , so on...

In [26]:

```

# if we set start_char with ord then it will keep updated.
num = int(input())
i = 1
while(i <= num):
    j = 1
    start_char = chr(ord('A') + i - 1)
    while(j <= num):
        char_target = chr(ord(start_char) + j - 1)
        print(char_target, end='')
        j = j + 1
    print()
    i = i + 1

```

```

4
ABCD
BCDE
CDEF
DEFG

```



In [ ]:

In [36]:

```
num = int(input())
i = 1
while(i<=num):
    j = 1
    print(" "*(num-i),end='')
    while(j<=i):

        print(j, end='')
        j = j + 1

    print()
    i = i + 1
```

```
5
  1
 12
123
1234
12345
```

In [ ]:

In [40]:

```
num = int(input())
i = 1
while(i<=num):
    j = 1
    print(" "*(num-i),end='')
    while(j<=i):

        print('*', end='')
        j = j + 1

    print()
    i = i + 1
```

```
3
 *
**
***
```

In [ ]: