**Day 2:**

**# pattern 1**

**#size 3**

----c----

--c-b-c--

c-b-a-b-c

--c-b-c--

----c----

**#size 5**

--------e--------

------e-d-e------

----e-d-c-d-e----

--e-d-c-b-c-d-e--

e-d-c-b-a-b-c-d-e

--e-d-c-b-c-d-e--

----e-d-c-d-e----

------e-d-e------

--------e--------

**Logic:**

size=3

letters=abc

columns=9 =((3\*2)-1)\*2)-1)

def rangoli(size):

alphabet="abcdefghijklmnopqrstuvwxyz"

letter=alphabet[:size][::-1] #cba

rows=[]

width=(((size\*2)-1)\*2)-1

for i in range(size):

row\_letter=letter[:i+1] #i=2 cb

row\_letter+=row\_letter[::-1][1:] #cbc

rows.append("-".join(row\_letter).center(width,"-")

rows=rows+rows[:size-1][::-1]

print("\n".join(rows))

**pattern 2:**

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \*

def pattern(size):

width=(((size\*2)-1)\*2)-1

rows=[]

for i in range(size):

row\_pattern="\*"\*i

rows.append(" ".join(row\_pattern).center(width," "))

print("\n".join(rows))

pattern(10)

**pattern 3:**

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

def pattern(size):

width=(((size\*2)-1)\*2)-1

rows=[]

for i in range(size):

row\_pattern="\*"\*i

rows.append(" ".join(row\_pattern).center(width," "))

rows+=rows[:size-1][::-1]

print("\n".join(rows))

pattern(10)

**pattern 4:**

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \*

\* \* \* \*

\* \* \*

\* \*

\*

def pattern(size):

width=(((size\*2)-1)\*2)-1

rows=[]

for i in range(size):

row\_pattern="\*"\*i

rows.append(" ".join(row\_pattern))

rows+=rows[:size-1][::-1]

print("\n".join(rows))

pattern(10)

**pattern 5:**

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \* \*

\* \* \* \* \* \* \* \*

\* \* \* \* \* \* \* \* \*

def pattern(size):

width=(((size\*2)-1)\*2)-1

rows=[]

for i in range(size):

row\_pattern="\*"\*i

rows.append(" ".join(row\_pattern))

#rows+=rows[:size-1][::-1]

print("\n".join(rows))

pattern(10)

**Pattern 6:**

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*

def pattern(size):

rows=[]

counter=size-1

for i in range(1,size+1):

for j in range(1,size+1):

if (i==j) and (counter!=0):

row\_pattern=" "\*counter

star="\*"\*i

final=row\_pattern+star

print(final)

counter-=1

pattern(11)

**Pattern 7:**

\*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\* \*

\*

def pattern(size):

width=(((size\*2)-1)\*2)-1

rows=[]

print("\*".center(width," "))

for i in range(size):

row\_pattern=" "\*i

final="\*"+row\_pattern+"\*"

rows.append(" ".join(final).center(width," "))

rows+=rows[:size-1][::-1]

print("\n".join(rows))

print("\*".center(width," "))

**Pattern 8:**

Sample input=9,27

------------.|.------------

---------.|..|..|.---------

------.|..|..|..|..|.------

---.|..|..|..|..|..|..|.---

----------WELCOME----------

---.|..|..|..|..|..|..|.---

------.|..|..|..|..|.------

---------.|..|..|.---------

------------.|.------------

# Enter your code here. Read input from STDIN. Print output to STDOUT

def doormat(n,m):

rows=[]

for i in range(n//2):

row\_pattern='.|.'\*((i\*2)+1)

rows.append("".join(row\_pattern).center(m,'-'))

rows.append("WELCOME".center(m,'-'))

rows+=(rows[:(n//2)][::-1])

print("\n".join(rows))

if \_\_name\_\_=='\_\_main\_\_':

n,m= map(int,str(input()).split(" "))

doormat(n,m)