In [2]:

```python
def bricks(N):
    m=N
    i=1
    while(N!=0):
        p=m-i
        if p<=0:
            print("Patlu")
            break
        else:
            m=p-(i*2)
            if m<=0:
                print("Motu")
                break
        i+=1
N=int(input())
bricks(N)
```

```
13
Motu
```

Regular Expressions

- pattern Matching
- symbolic notation of a pattern

```
    - pattern:Format which Repeats
    - Pattern(RE):Represent The set of all values which matches a patt
  ern
```

- [0-9]--->Any digit
- [a-z]--->Any lower case alphabet
- [2468] or [6428] or[8642] or [any order]--->All single digit multiple of 2-
- ^[0-9]{1}$--->only represent single digit numbers
- ^[0-9]{3}$--->only represent three digit numbers
- ^[0-9]{4}$--->only represent four digit numbers
- ^[1-9][0-9]*0$--->all multiples of 10
- ^([0-9][1-9]*[50])|([5]) ->All multiples of 5
- ^[0-9]{10}$ -->All 10 digit numbers
- ^[9876][0-9]{9}$ -->All the contact numbers start with [9876] and it contains 9 numbers
- [p][r][i][n][t] or (print)---> Searching for a word
- Email validation(username@domain.extension (mailto:username@domain.extension))

- User name

```
Length of username : [6,15]

No special characters other than _  .

it should not begin and end with _ .

character set:all digits and lower case alphabets

^[0-9a-z][0-9a-z_.]{4,15}[0-9a-z]$
```

- domain

```
Length of Domain : [3,18]

No Special character

character set: it should be lower case characters and all digits

^[a-z0-9]{3,18}[a-z0-9]$
```

- Extension

```
Length of extension : [2,4]

No Special character

character set: it should be lower case alphabets only

^[a-z0-9]{2,4}[a-z0-9]$
```

-- (^[0-9a-z][0-9a-z_.]{4,15}[0-9a-z][@][a-z0-9]{3,18}[a-z0-9][.][a-z0-9]{2,4}[a-z0-9]$) # format for

username@domain.extension (mailto:username@domain.extension)

- (^[0-9a-z][0-9a-z_.]{4,15}[0-9a-z][@][a-z0-9]{3,18}[a-z0-9][.][a-z0-9]{2,4}[a-z0-9][.][a-z0-9]{2,4}[a-z0-9]) # format for

username@domain.extension.edu (mailto:username@domain.extension.edu)

- ^[a].*[z]$ -->Any string of any length starts with a and stops to s

In [32]:

```
import re
def emailvalidator(email):
    pattern="(^[0-9a-z][0-9a-z_.]{4,15}[0-9a-z][@][a-z0-9]{3,18}[a-z0-9][.][
    if re.match(pattern,email):
        return True
    return False
emailvalidator("alekhyaganji440@gmail.com")
```

Out[32]:

```
True
```

In [19]:

```
#Function to validate a phone number
import re

def phonenumbervalidator(number):
    pattern = "^[9876][0-9]{9}$|^[0][6-9][0-9]{9}$|^[+][9][1][6-9][0-9]{9}$"
    if re.match(pattern,str(number)):
        print("Valid Number")
    else:
        print("Invalid Number")

phonenumbervalidator("+919010203749")


```

```
Valid Number
```

Out[19]:

```
True
```

In [38]:

```
#New contacts in a given dictionary
#Merge two data
def importcontact(newcontacts):
    contacts.update(newcontacts)
    print(len(newcontacts.keys()),"contacts added successfully")
    return
newcontacts={"nam1":1344444,"na2":13444}
importcontact(newcontacts)
newcontacts.items()
```

```
2 contacts added successfully
```

Out[38]:

```
dict_items([('nam1', 1344444), ('na2', 13444)])
```

In [40]:

```
contacts={"name1":[9876543210,"name1@domain.ext"]}
def addcontact(name,phone,email):
    if name in contacts:
        print(name,"already exists.")
    else:
        if not phonenumbervalidator(phone):
            print("Invalid phone number")
            return
        if not emailvalidator(email):
            print("Invalid email address")
            return
        newcontact=[]
        newcontact.append(phone)
        newcontact.append(email)
        contacts[name]=newcontact
    return
addcontact("alekhya",8328363233,'alekhyaganji440@gmail.com')
```

```
Valid Number
Invalid phone number
```

In [31]:

```
def searchcontact(name):
    if name  in contacts:
        print(name)
        print("phone:",contacts[name][0])
        print("email :",contacts[name][0])
    else:
        print("%s does not exist"% name)
    return
searchcontact("name1")
```

```
name1
phone: 9876543210
email : 9876543210
```

In [35]:

```
contacts={"a":"123144","b":"7518"}
def listofcontacts():
    for key,value in contacts.items():

        print(key,":",value)
listofcontacts()
```

```
a : 123144
b : 7518
```

In [46]:

```
contacts={"name1":[9876543210,"name1@domain.ext"]}
def listofcontancts():
    for contact,info in contact.items():
        print(contact,"\n","phone :"info[0],"\n","email :"info[0])
listofcontacts()
```

```
  File "<ipython-input-46-b5ffaf5c7c29>", line 4
    print(contact,"\n","phone :"info[0],"\n","email :"info[0])
                                   ^
SyntaxError: invalid syntax
```

In [ ]:

```

```

File Handling in Python

- File - Document containing information residing on the permanent storage
- Types -Text,Pdf,Csv etc
- File I/O - Channeling I/O data to files
- Default I/O Channels - keyboard /screen
- Change I/O Channel to files for Reading and Writing
- Read a file - Input from file
- Write to a file - Output to a file
- Read/Write a file - open(filename,mode)

In [ ]:

```

```

In [60]:

```
#Function to read a File
def readFile(filename):
    s=open(filename,'r')
    filedata=s.read()
    s.close()
    return filedata
filename='DataFiles/data.txt'
filedata=readFile(filename)
#for line in filedata.split('\n'):
#    print (line)

def printdatalines(filename):
    f=open(filename,'r')
    for line in f:
        print(line,end="")
    return
printdatalines(filename)
```

```
Line1
Line2
Line3
```

In [1]:

```
#Function to write data into a file

def writeintofile(filename,filedata):
    with open(filename,'w') as f:
        f.write(filedata)
    return
filename='DataFiles/data.txt'
writeintofile(filename,"newdata\n")
```

In [2]:

```
#Function to append data

def appendthedata(filename,filedata):
    with open(filename,'a') as f:
        f.write(filedata)
    return
filedata= "Line2\nLine3"
appendthedata(filename,filedata)
```

In [1]:

```
#Function to add contact to contacts text file
from Packagess.validations import phonenumbervalidator as pnv, emailvalidato

def addContact(name,number,email):
    # store data as name,phonenum,email in the contacts text file
    filename = 'DataFiles/contact.txt'
    if not validate(name):
        if pnv(number)and ev(email):
            with open(filename,'a') as f:
                line = name + ',' +str(number) + ',' + email + '\n'
                f.write(line)
            print(name,"Added to contacts")
        else:
            print("Invalid Phone Number or email")
            return
    else:
        print(name,"already exists")
    return
# Function to check if contact already exists
import re
def validate(name):
    filename='DataFiles/contact.txt'
    with open(filename,'r') as f:
        filedata = f.read()
        pattern = name+','
    return re.search(pattern,filedata)
addContact('name1',7890645321,'abcd_123@yahoo.com')

```

```
name1 Added to contacts
```

In [10]:

```
#Function from csvfile to list
filename = 'DataFiles/contact.txt'
def csvFileToList(filename):
    li=[]
    with open(filename,'r') as f:
        for line in f:
            li.append(line.split(','))
    return li
csvFileToList(filename)

```

Out[10]:

```
[['name1', '7890645321', 'abcd_123@yahoo.com\n'],
 ['name2', '9010203040', 'alekhya_13@gmail.com\n']]
```

In [11]:

```
#FUnction from list to csvfile
def listToFile(li):
    s = ''
    for i in li:
        s+=','.join(i)
    return s
li=csvFileToList(filename)
listToFile(li)
```

Out[11]:

```
'name1,7890645321,abcd_123@yahoo.com\nname2,9010203040,alekhya_13@gmail.com\n'
```

In [4]:

```
#Function to add contact to contacts text file
from Packagess.validations import phonenumbervalidator as pnv, emailvalidato

def addContact(name,number,email):
    # store data as name,phonenum,email in the contacts text file
    filename = 'DataFiles/contact.txt'
    if not validate(name):
        if pnv(number)and ev(email):
            with open(filename,'a') as f:
                line = name + ',' +str(number) + ',' + email + '\n'
                f.write(line)
            print(name,"Added to contacts")
        else:
            print("Invalid Phone Number or email")
            return
    else:
        print(name,"already exists")
    return
# Function to check if contact already exists
import re
def validate(name):
    filename='DataFiles/contact.txt'
    with open(filename,'r') as f:
        filedata = f.read()
        pattern = name+','
    return re.search(pattern,filedata)
addContact('name2',9010203040,'alekhya_13@gmail.com')
```

```
name2 Added to contacts
```

In [1]:

```python
#Function to update the data
from Packagess.validations import phonenumbervalidator as pnv, emailvalidato

def addContact(name,number,email):
    # store data as name,phonenum,email in the contacts text file
    filename = 'DataFiles/contact.txt'
    if not validate(name):
        if pnv(number)and ev(email):
            with open(filename,'a') as f:
                line = name + ',' +str(number) + ',' + email + '\n'
                f.write(line)
            print(name,"Added to contacts")
        else:
            print("Invalid Phone Number or email")
            return
    else:
        print(name,"already exists")
    return
# Function to check if contact already exists
import re
def validate(name):
    filename='DataFiles/contact.txt'
    with open(filename,'r') as f:
        filedata = f.read()
        pattern = name+','
    return re.search(pattern,filedata)
addContact('name2',9010203040,'alekhya_13@gmail.com')
```

```
name2 already exists
```

In [8]:

```python
import re
s="hiiabbb"
pattern='[a]b{2,3}'
k=re.findall(pattern,s)
print(k)

```

```
['abbb']
```

In [17]:

```python
a=int(input())
b=int(input())
sum=0
for i in range(a,b+1):
    sum=sum+i
    avg=sum//b
print(avg)
```

```
1
3
2
```

In [ ]:

```python

```