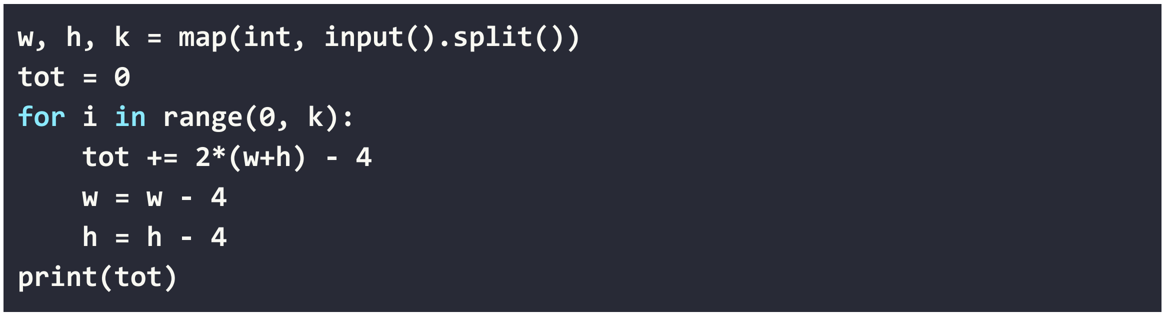
CodeForces

1. 1031A - Golden Plate

Simple pattern finding would get the solve and adding those progressions for each iteration would give the final result.

Code snippet :



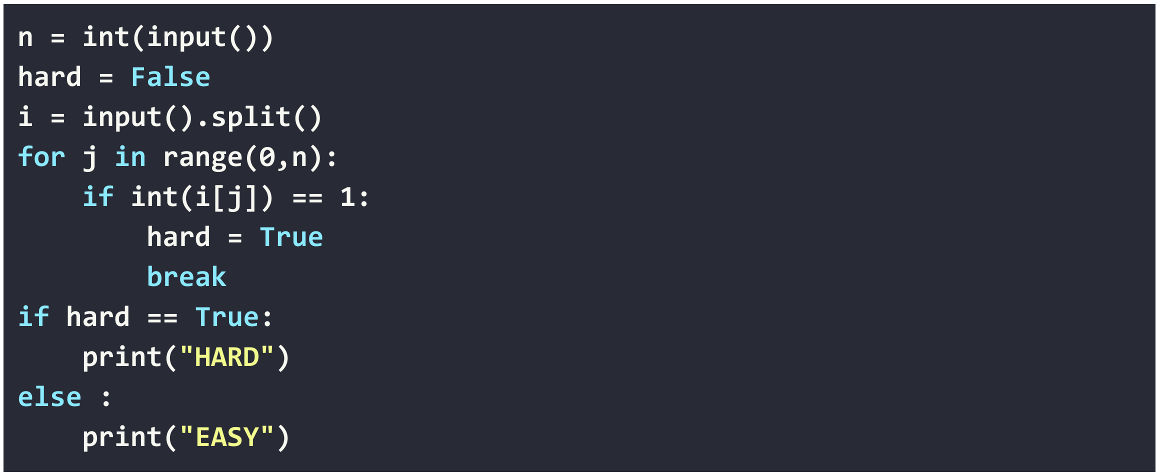
```
w, h, k = map(int, input().split())
tot = 0
for i in range(0, k):
    tot += 2*(w+h) - 4
    w = w - 4
    h = h - 4
print(tot)
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/1031A.py

2. 1030A - In Search of an Easy Problem

Check if any of the input contains the number 1, if it does set "hard" flag as "True" and break and print "HARD" else if input does not contain the number 1, "hard" flag would be "False" and thus print "EASY".

Code snippet :



```
n = int(input())
hard = False
i = input().split()
for j in range(0,n):
    if int(i[j]) == 1:
        hard = True
        break
if hard == True:
    print("HARD")
else :
    print("EASY")
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/1030A.py

3. 1095A - Repeating Cipher

After getting the input and converting it to a list, we try to print all the characters in, 0, 2, 5, 9 ... format to print out characters until an exception occurs.

Code snippet :

```
n = int(input())
s = list(input())
j = 0
for i in range(1,n+1):
    j += i
    try:
        print(s[j-1], end='')
    except:
        break
print()
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/1095A.py

4. 1102A - Integer Sequence Dividing

Trying out multiple examples gave in a pattern that all inputted numbers which are divisible by 4 or get a remainder of 3 when dividing by 4 has the minimum difference as 0 and everything else has the minimum difference as 1.

Code snippet :

```
n = int(input())
if n%4 == 0 or n%4 == 3:
    print(0)
else:
    print(1)
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/1102A.py

5. 1061A - Coins

Coins of maximum value are added until it exceeds the required sum, thus the number of times the coins added gives the answer.

In case, the sum is less than the maximum value of a coin then that means there is a coin with a value equal to sum in the given set thus making count as 1.

Code snippet :

```
n, s = map(int, input().split())
count = 1
tmp = int(n)
while s > tmp:
    count+=1
    tmp+=n
print(count)
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/1061A.py

6. 1064A - Make a triangle!

After a little bit of research I found out that a triangle is said to have a "positive area" when the sum of two of its least sides is greater than the largest side.

After checking the tutorial section of the challenge also confirmed this.

Thus, I increment the least side alternatively until it becomes greater than or equal to the third greater side. Technically, a bruteforce method.

Code snippet :

```
lst = list(map(int, input().split()))
lst.sort()
i = -1
count = 0
while True:
    if lst[0] + lst[1] - 1 >= lst[2]:
        break
    else:
        if i < 0:
            lst[0]+=1
        else:
            lst[1]+=1
        count+=1
        i*=(-1)
print(count)
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/1064A.py

7. 1096A - Find Divisible

Basic math implementation for each input.

Code snippet :

```
t = int(input())
res = []
for i in range(0, t):
    l, r = map(int, input().split())
    x = l
    y = r - (r%x)
    res.append(f'{x} {y}')
for i in res:
    print(i)
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/1096A.py

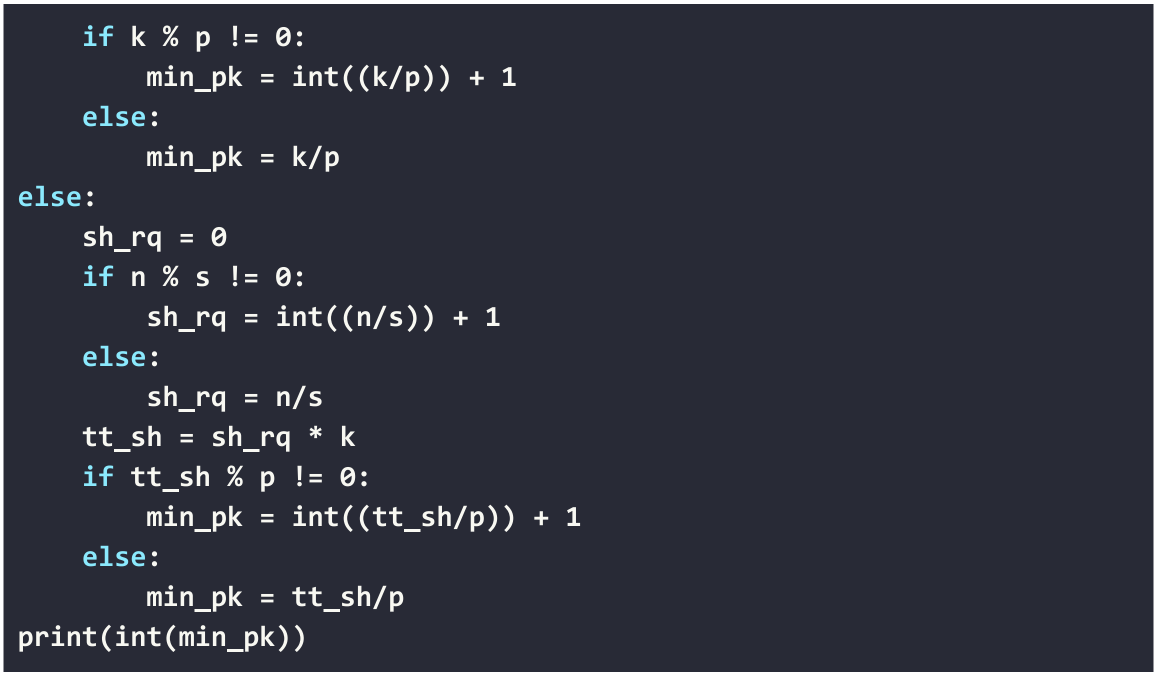
8. 965A - Paper Airplanes

If the number of airplanes that can be made from a single sheet is greater than the number of airplanes each person can make then each person requires only one packet.

If the no of airplanes that can be made from a single sheet is lesser than the no of airplanes each person can make then one person requires (no of airplanes each can make/no of airplanes from one sheet) no of sheets.

Code snippet :

```
k, n, s, p = map(int, input().split())
min_pk = 0
if s > n:
```



```
    if k % p != 0:
        min_pk = int((k/p)) + 1
    else:
        min_pk = k/p
else:
    sh_rq = 0
    if n % s != 0:
        sh_rq = int((n/s)) + 1
    else:
        sh_rq = n/s
    tt_sh = sh_rq * k
    if tt_sh % p != 0:
        min_pk = int((tt_sh/p)) + 1
    else:
        min_pk = tt_sh/p
print(int(min_pk))
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/965A.py

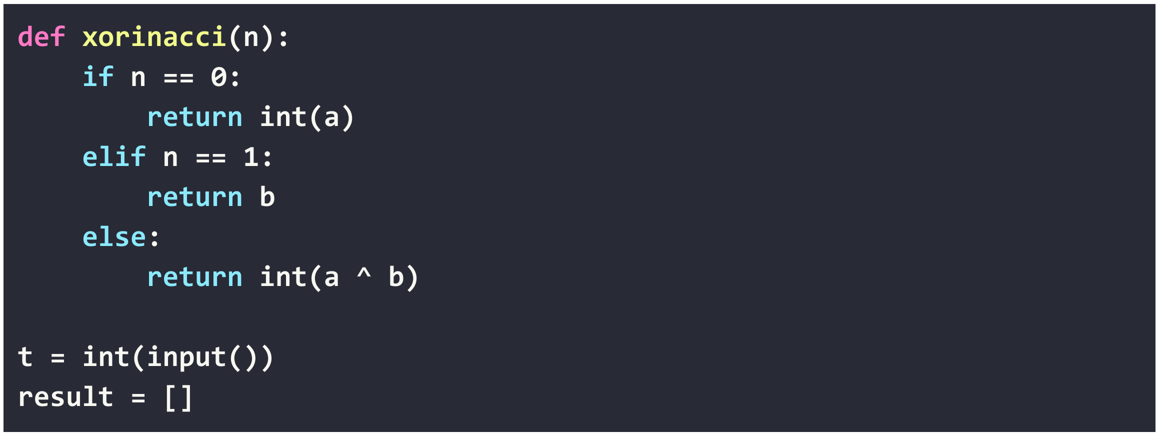
9. 1208A - XORinacci

First I tried a recursion function method but since few inputs exceeded the recursion limit I needed another approach.

With a little bit of research and trying various inputs a pattern emerged that every third input will have either 1 or 0 or any other number as a remainder when divided by 3.

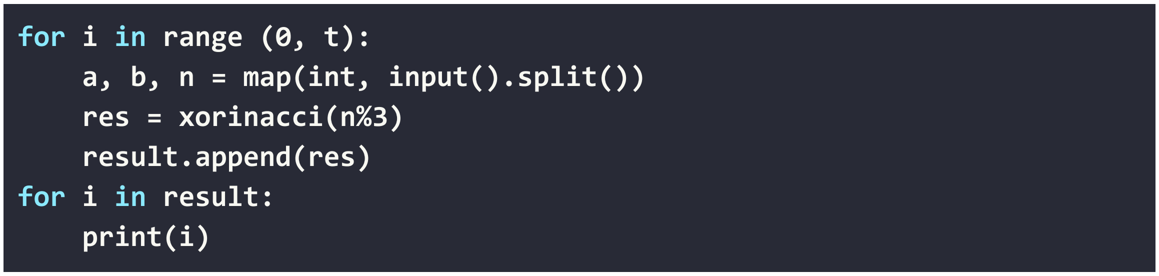
And if it is indeed any other number then the answer is always xor of the first two inputs.

Code snippet :



```
def xorinacci(n):
    if n == 0:
        return int(a)
    elif n == 1:
        return b
    else:
        return int(a ^ b)

t = int(input())
result = []
```



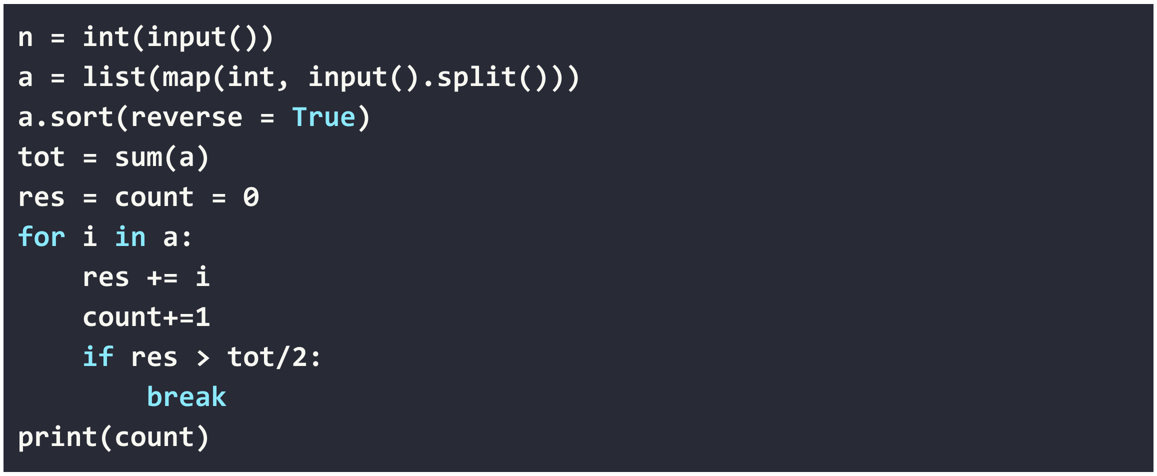
```python
for i in range (0, t):
    a, b, n = map(int, input().split())
    res = xorinacci(n%3)
    result.append(res)
for i in result:
    print(i)
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/1208A.py

10. 160A - Twins

After a little bit of research I found out that sorting the list and adding the larger elements until the sum becomes greater and keeping a counter for each iteration over the list would give the minimum no of coins required.

Code snippet :



```python
n = int(input())
a = list(map(int, input().split()))
a.sort(reverse = True)
tot = sum(a)
res = count = 0
for i in a:
    res += i
    count+=1
    if res > tot/2:
        break
print(count)
```

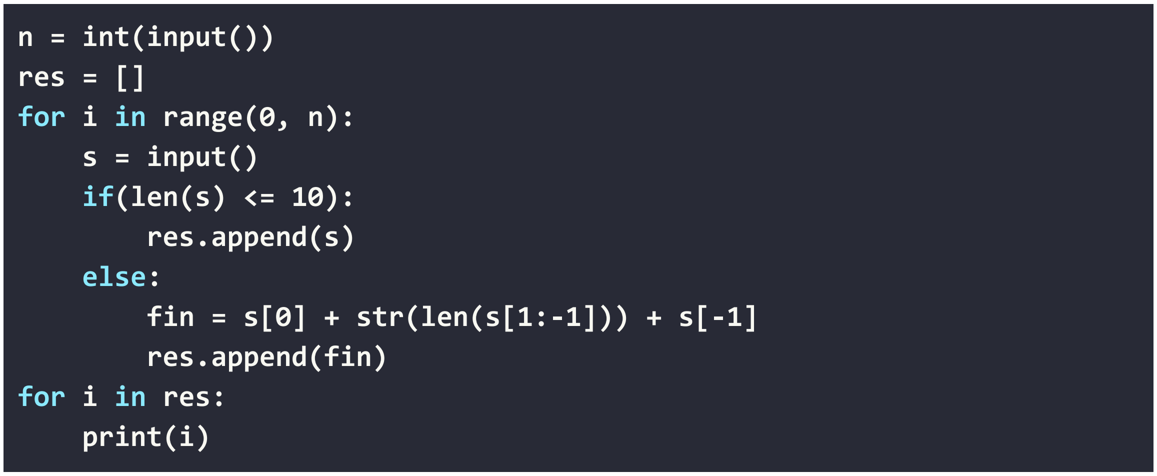
Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/160A.py

11. 71A - Way too long words

If the length of the inputted word is less than or equal to 10 then the result would be the word itself.

Since the solution is in python it is easy to implement, the final result would be the first character of the inputted string concat with the length between the first and the last characters(using "len(s[1:-1])") concat with the last character.

Code snippet :



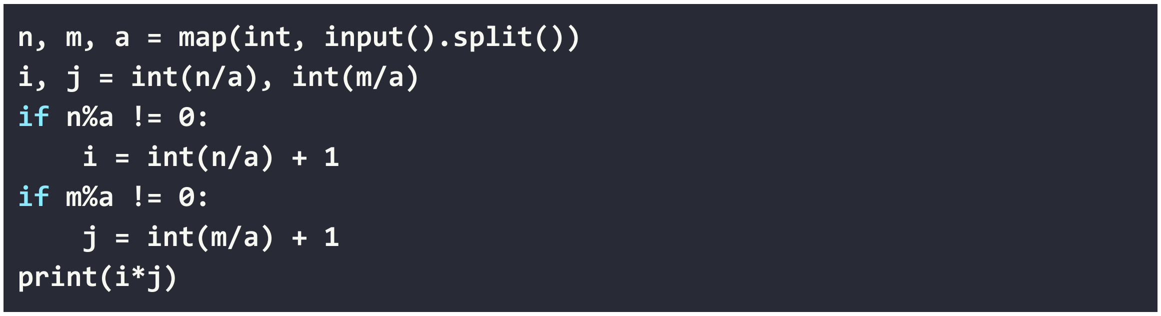
```
n = int(input())
res = []
for i in range(0, n):
    s = input()
    if(len(s) <= 10):
        res.append(s)
    else:
        fin = s[0] + str(len(s[1:-1])) + s[-1]
        res.append(fin)
for i in res:
    print(i)
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/71A.py

12. 1A - Theatre Square

Dividing the side length of the square with the stone on both the sides and multiplying them further would give the total no of stones needed. But if it leaves a remainder and since the stone can't be cut in half we could round the quotient to the next number.

Code snippet :



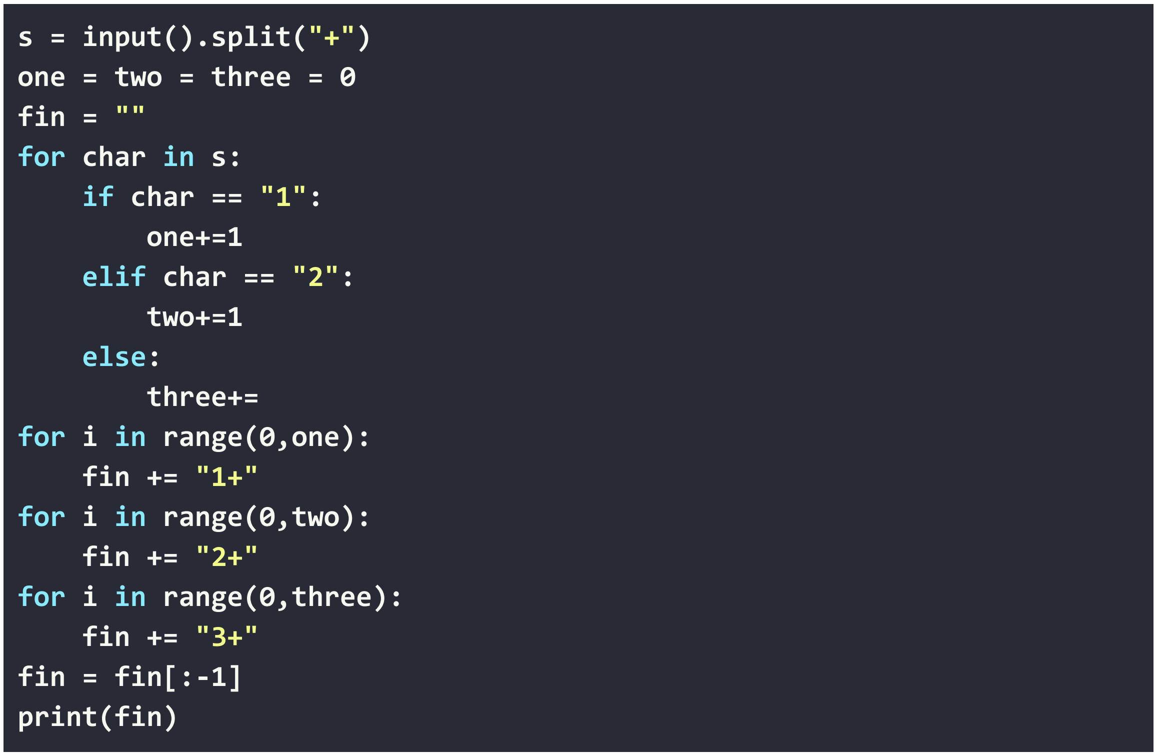
```
n, m, a = map(int, input().split())
i, j = int(n/a), int(m/a)
if n%a != 0:
    i = int(n/a) + 1
if m%a != 0:
    j = int(m/a) + 1
print(i*j)
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/1A.py

13. 339A - Helpful Maths

Count the number of one, two and three and use three different loops concat "1+", "2+", "3+" respectively for the range of the count.

Code snippet :



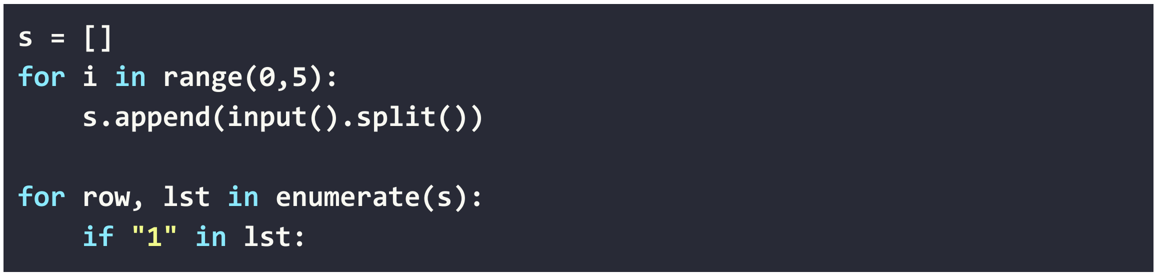
```
s = input().split("+")
one = two = three = 0
fin = ""
for char in s:
    if char == "1":
        one+=1
    elif char == "2":
        two+=1
    else:
        three+=
for i in range(0,one):
    fin += "1+"
for i in range(0,two):
    fin += "2+"
for i in range(0,three):
    fin += "3+"
fin = fin[:-1]
print(fin)
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/339A.py

14. 263A - Beautiful Matrix

The shortest distance to the center index is the difference between the current index and the center.

Code snippet :



```
s = []
for i in range(0,5):
    s.append(input().split())

for row, lst in enumerate(s):
    if "1" in lst:
        col =  lst.index("1")
        break
fin_row = abs(row-2)
fin_col = abs(col-2)
print(fin_col+fin_row)
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/263A.py

15. 705A - Hulk

Setting different parts of the string in different variables and alternating the string to concat for every iteration and also removing the "that" string with "it" finally would give the final result required.

Code snippet :

```
n = int(input())
s1 = "I hate "
s2 = "I love "
that = "that "
it = "it "
res = ""
if n == 1:
    res = s1 + it
else:
    for i in range(0, n):
        if i%2 == 0:
            res += s1 + that
        else:
            res += s2 + that
    res = res[:-5] + it
print(res)
```

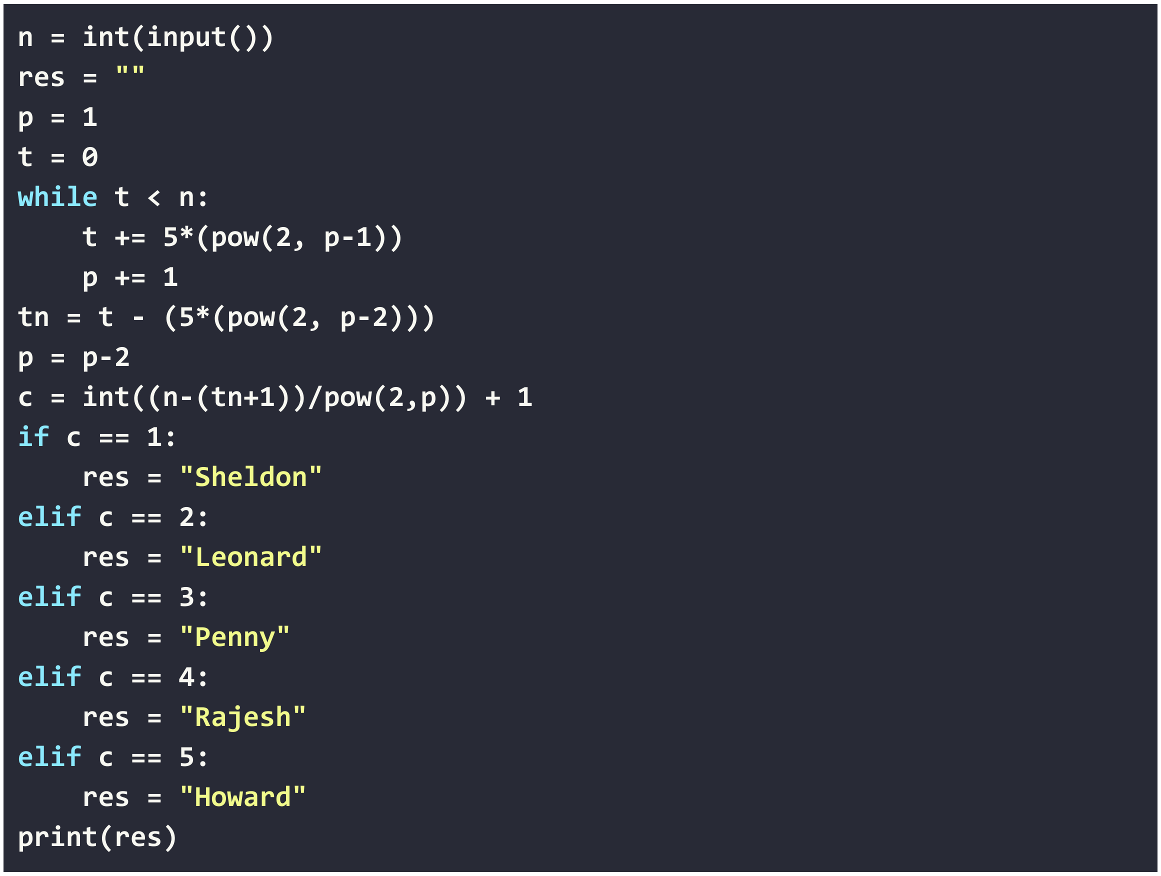
Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/705A.py

16. 82A - Double Cola

After a lot of trial and error and codeforces announcements, I managed to find a solution which finds the next G.P. value lesser than to the given input and add the no of iterations to find the answer.

But, Time Limit was exceeded. Then from a forum where this problem was discussed there was a hint given to divide the difference between the result I got and the input by the no of iterations, which drastically reduced the time taken by making my former solutions' second loop more easier.

Code snippet :



```
n = int(input())
res = ""
p = 1
t = 0
while t < n:
    t += 5*(pow(2, p-1))
    p += 1
tn = t - (5*(pow(2, p-2)))
p = p-2
c = int((n-(tn+1))/pow(2,p)) + 1
if c == 1:
    res = "Sheldon"
elif c == 2:
    res = "Leonard"
elif c == 3:
    res = "Penny"
elif c == 4:
    res = "Rajesh"
elif c == 5:
    res = "Howard"
print(res)
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/82A.py

17. 96A - Football

Two counters, "zer" and "one" is set to 0 initially and during each iteration over every character in the list the counter is increased by 1 as long as that character is the same for every next iteration, in case the character changes the counter is set to 0 again.

Code snippet :

```
s = list(input())
zer = one = 0
found = False
for char in s:
    if char == "0":
        if one != "0":
            one=0
        zer+=1
    elif char == "1":
        if zer != "0":
            zer=0
        one+=1
    if zer == 7 or one == 7:
        found = True
if found != True:
    print("NO")
elif found == True:
    print("YES")
```

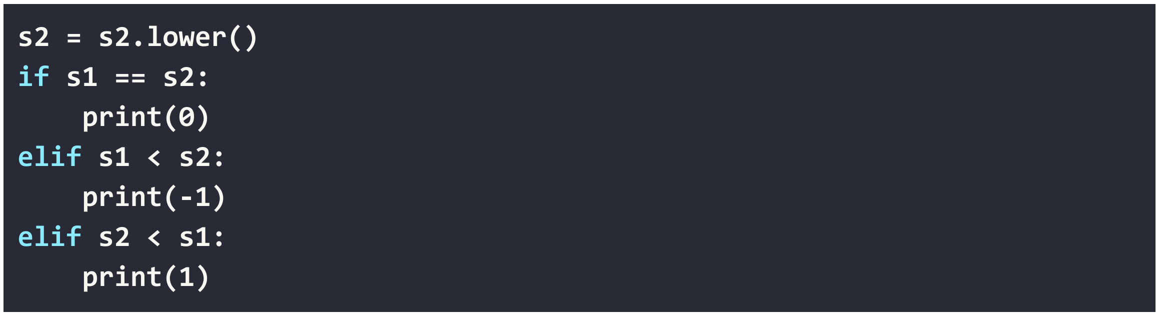
Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/96A.py

18. 112A - Petya and Strings

All strings are integers at the fundamental level and since comparison between them is done built-in in all programming languages a simple if/else or switch/case would only require.

Code snippet :

```
s1 = input()
s2 = input()
s1 = s1.lower()
```



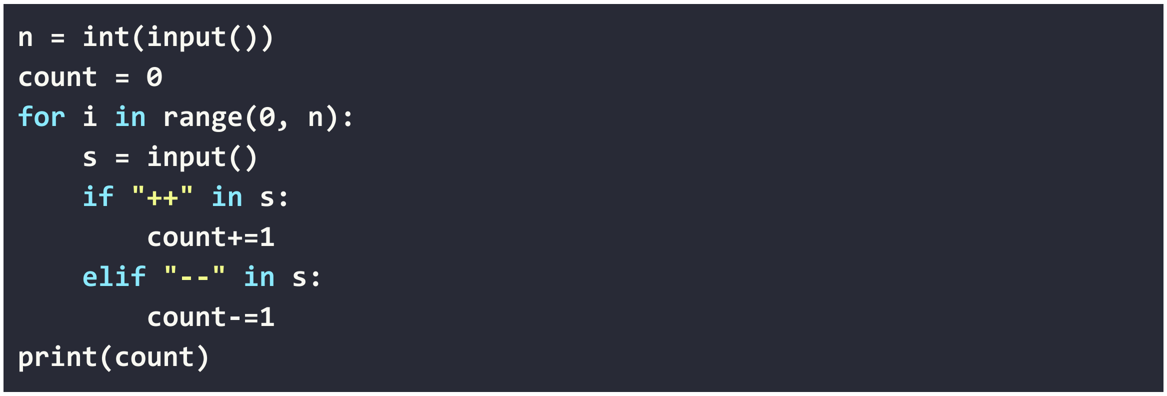
```
s2 = s2.lower()
if s1 == s2:
    print(0)
elif s1 < s2:
    print(-1)
elif s2 < s1:
    print(1)
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/112A.py

19. 282A - Bit++

If "++" is in the input the counter increases by one and else the counter decreases by one, thus the total value can be found after all the inputs.

Code snippet :



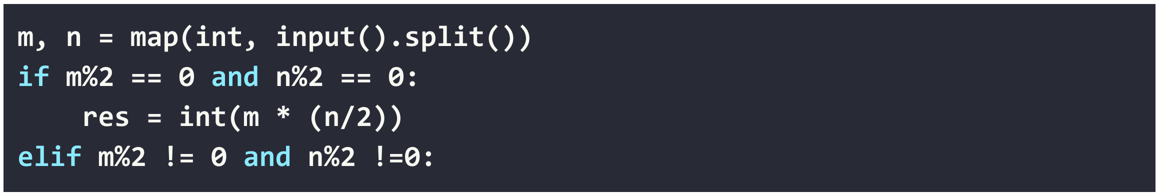
```
n = int(input())
count = 0
for i in range(0, n):
    s = input()
    if "++" in s:
        count+=1
    elif "--" in s:
        count-=1
print(count)
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/282A.py

20. 50A - Domino piling

Simple area calculation, of sides respective of whether the side is even or odd.

Code snippet :



```
m, n = map(int, input().split())
if m%2 == 0 and n%2 == 0:
    res = int(m * (n/2))
elif m%2 != 0 and n%2 !=0:
    res = (m * int(n/2)) + int(m/2)
elif m%2 != 0 and n%2 == 0:
    res = int(m * (n/2))
else:
    res = int(n * (m/2))
print(res)
```

Full code at :
https://github.com/AmunRha/ChallengeSet1/blob/master/Codeforces/50A.py