Day 0

To complete this challenge, you must save a line of input from stdin to a variable, print Hello, World. on a single line, and finally print the value of your variable on a second line.

In [3]:

```
inputString = input()

print('Hello, World.')
print(inputString)
```

```
This is Calvin
Hello, World.
This is Calvin
```

DAY 1

Complete the code in the editor below. The variables , , and are already declared and initialized for you. You must:

Declare variables: one of type int, one of type double, and one of type String. Read lines of input from stdin (according to the sequence given in the Input Format section below) and initialize your variables. Use the operator to perform the following operations: Print the sum of plus your int variable on a new line. Print the sum of plus your double variable to a scale of one decimal place on a new line. Concatenate with the string you read as input and print the result on a new line.

In [1]:

```
l = 2
d = 4.5
s = 'Hello World'

int1 = int(input('Enter Integer: '))
double1 = float(input('Enter Double: '))
string1 = str(input('Enter String: '))

print(l+ int1)
print(d+ double1)
print(s +string1)
```

```
Enter Integer: 10
Enter Double: 5.2
Enter String: This is Day 1
12
9.7
Hello WorldThis is Day 1
```

Day 2: Operators

Given the meal price (base cost of a meal), tip percent (the percentage of the meal price being added as tip), and tax percent (the percentage of the meal price being added as tax) for a meal, find and print the meal's total cost. Round the result to the nearest integer.

In [4]:

```
meal = float(input("Enter Meal Cost: "))
tip = float(input("Enter Tip percent: "))
tax = float(input("Enter Tax percent: "))
total = round(meal*(100+tip+tax)/100)
print(total)
```

```
Enter Meal Cost: 100
Enter Tip percent: 0.5
Enter Tax percent: 0.02
101
```

Day 3: Intro to Conditional Statements

Given an integer, n, perform the following conditional actions: If n is odd, print Weird If n is even and in the inclusive range of 2 to 5, print Not Weird If n is even and in the inclusive range of 6 to 20, print Weird If n is even and greater than 20, print Not Weird Complete the stub code provided in your editor to print whether or not is weird.

In [7]:

```
n = int(input('enter integer: '))
if n % 2 == 1 or n in range(6, 21, 2):
 print('Weird')
else:
 print('Not Weird')
```

```
enter integer: 22
Not Weird
```

Day 4: Class vs. Instance

Write a Person class with an instance variable, age, and a constructor that takes an integer, initialAge, as a parameter. The constructor must assign initialAge to Age after confirming the argument passed as initialAge is not negative; if a negative argument is passed as initialAge, the constructor should set age to 0 and print Age is not valid, setting age to 0.. In addition, you must write the following instance methods: yearPasses() should increase the instance variable by 1. amIOld() should perform the following conditional actions: If , print You are young.. If and , print You are a teenager.. Otherwise, print You are old.. To help you learn by example and complete this challenge, much of the code is provided for you, but you'll be writing everything in the future. The code that creates each instance of your Person class is in the main method. Don't worry if you don't understand it all quite yet!

In [4]:

```
class Person:

    def __init__(self, initial_age=0):
        if initial_age < 0:
            print("Age is not valid, setting age to 0.")
            inititial_age = 0
        self.age = initial_age

    def yearPasses(self):
        self.age += 1

    def amIOld(self):
        if self.age < 13:
            print("You are young")
        elif self.age < 18:
            print("You are a teenager")
        else:
            print("You are old")
for t in range(3):
    initial_age = int(input('Enter Age: '))
    person = Person(initial_age=initial_age)
    person.amIOld()
    person.yearPasses()
    person.amIOld()
```

```
Enter Age: 12
You are young
You are a teenager
Enter Age: -2
Age is not valid, setting age to 0.
```

```
You are young
You are young
Enter Age: 17
You are a teenager
You are old
```

Day 5:

Given an integer, n, print its first 10 multiples.

In [6]:

```
x = int(input("enter integer: "))
for i in range(1,11):
    result = x*i
    print("{} x {} = {}".format(x,i,result))
```

```
enter integer: 12
12 x 1 = 12
12 x 2 = 24
12 x 3 = 36
12 x 4 = 48
12 x 5 = 60
12 x 6 = 72
12 x 7 = 84
12 x 8 = 96
12 x 9 = 108
12 x 10 = 120
```

Day 6:

Given a string, s, of length N that is indexed from 0 to N-1, print its even-indexed and odd-indexed characters as 2 space-separated strings on a single line (see the Sample below for more detail).

In [9]:

```
input_string = input('Enter string word: ')
first_half = [input_string[i] for i in range(0,len(input_string),2)]
second_half = [input_string[i] for i in range(1,len(input_string),2)]
print(first_half,'   ',second_half)
```

```
Enter string word: helloworld
['h', 'l', 'o', 'o', 'l']     ['e', 'l', 'w', 'r', 'd']
```

Day 7:

Given an array, A, of N integers, print A's elements in reverse order as a single line of space-separated numbers.

In [15]:

```
A = [1,2,3,4,5,6,7,8,9,10]
A = A[::-1]
for i in range(0,len(A)):
    print(A[i],end = ' ')
```

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

Day 8:

Given n names and phone numbers, assemble a phone book that maps friends' names to their respective phone numbers. You will then be given an unknown number of names to query your phone book for. For each name queried, print the associated entry from your phone book on a new line in the form name=phoneNumber; if an entry for name is not found, print Not found instead.

In [2]:

```
length = int(input("Enter length of Phonebook:"))
phonebook = {}
for i in range(length):
    data = input('Enter name and number')
    data = data.split()
    phonebook[data[0]] = data[1]
phonebook
```

```
Enter length of Phonebook:4
Enter name and numberJamie 12345
Enter name and numberOliver 98765
Enter name and numberRobert 314159
Enter name and numberDowney 1433000
```

Out[2]:

```
{'Jamie': '12345', 'Oliver': '98765', 'Robert': '314159', 'Downey': '1433000'}
```

In [3]:

```
search = ''
while search!='quit':
    search = input('Enter query: ')
    try:
        print("{} - {}".format(search,phonebook[search]))
    except:
        print('Entry Not Found')
```

```
Enter query: Jamie
Jamie - 12345
Enter query: GordonRamsay
Entry Not Found
Enter query: Robert
Robert - 314159
Enter query: quit
Entry Not Found
```

In [28]:

```
phonebook
```

Out[28]:

```
{'J': 'a', 'C': 'h', 'R': 'o', 'D': 'o'}
```

In [29]:

```
'an apple'.split()
```

Out[29]:

```
['an', 'apple']
```

Day 9:

Recursive Method for Calculating Factorial

In [7]:

```
def factorial(n):
    if n ==0:
        return 1
    else:
        return n*factorial(n-1)
x = int(input('Enter number: '))
print(factorial(x))
```

```
Enter number: 6
720
```

Day 10:

Given a base-10 integer, n, convert it to binary (base-2). Then find and print the base-10 integer denoting the maximum number of consecutive 1's in n's binary representation. When working with different bases, it is common to show the base as a subscript.

In [9]:

```
def max(a,b):
    return a if a>b else b
n = int(input())

max_num = 0
i = 0

while n:
    while n&1:
        i+=1
        n>>=1
    max_num = max(i, max_num)
    if not n&1:
        count = 0
        n>>=1

print(max_num)
```

```
7
3
```

Day 11:

Calculate the hourglass sum for every hourglass in A, then print the maximum hourglass sum.

In [12]:

```
array= []
for array_i in range(6):
    array_temp = list(map(int,input().strip().split(' ')))
    array.append(array_temp)
max = 0

for i in range(0,4):
    for j in range(0,4):
        sum = 0
        sum= array[i][j]+array[i][j+1]+array[i][j+2]+array[i+1][j+1]+array[i+2][j]+arra
y[i+2][j+1]+array[i+2][j+2]
        if i==0 and j==0:
            max = sum
        if sum > max:
            max =sum

print(max)
```

```
1 1 1 0 0 0
0 1 0 0 0 0
1 1 1 0 0 0
0 0 2 4 4 0
0 0 0 2 0 0
0 0 1 2 4 0
19
```

Day 12: Inheritence

You are given two classes, Person and Student, where Person is the base class and Student is the derived class. Completed code for Person and a declaration for Student are provided for you in the editor. Observe that Student inherits all the properties of Person.

In [22]:

```
class Student():
    def __init__(self,f_name,l_name,st_id,scores,n):
        self.scores=scores
        self.n = n
    def avg_scores(self):
        avg = 0.0
        for score in self.scores:
            avg += score
            avg= avg/len(self.scores)
            if avg<40:
                return "T"
            elif avg< 55:
                return 'D'
            elif avg<70:
                return'P'
            elif avg<80:
                return'A'
            elif avg<90:
                return'E'
            else:
                return 'O'
    def print_st(self):
        print(f"Name: {f_name}")
        print(f"ID: {st_id}")
        print("Grade:",end=" ")
        print(self.avg_scores())

f_name=input("Enter your first name: ")
l_name=input("Enter your last name: ")
st_id=int(input("Enter your student ID: "))
n=int(input("Enter number of scores to enter: "))
scores=[]

for i in range(n):
    scores_inp=int(input("Enter the score: "))
    scores.append(scores_inp)
    st=Student(f_name,l_name,st_id,scores,n)
    st.print_st()
```

```
Enter your first name: Calvin
Enter your last name: Pinto
Enter your student ID: 70091019052
Enter number of scores to enter: 4
Enter the score: 82
Name: Calvin
ID: 70091019052
Grade: E
Enter the score: 79
Name: Calvin
ID: 70091019052
Grade: D
Enter the score: 99
Name: Calvin
ID: 70091019052
Grade: T
Enter the score: 60
Name: Calvin
ID: 70091019052
Grade: T
```

In [26]:

```
```

```
Name: Calvin Pinto
ID: 7009101

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-26-e874efefe390> in <module>
     17 print(f'Name: {s1.first_name} {s1.last_name}')
     17 print(f'Name: {s1.first_name} {s1.last_name}')
     18 print(f'ID: {s1.id_number}')
---> 19 print(f'Grade: {s1.calculate()}')

<ipython-input-26-e874efefe390> in calculate(self)
      9         d = {'T': 40,'D': 55,'P': 70,'A': 80,'E': 90,'O': 100,}
     10
---> 11         avg = sum(self.scores) / len(self.scores)
     12
     13         for grade, score_threshold in d.items():

TypeError: 'int' object is not callable
```

Day 13

In [30]:

```
class my_library():
    def __init__(self,title,author,price):
        self.title=title
        self.author=author
        self.price=price
    def print_library(self):
        print(f"Title: {self.title}")
        print(f"Author: {self.author}")
        print(f"Price: {self.price}")
lib = my_library('Harry Porter','Master Jones',500)
lib.print_library()
```

```
Title: Harry Porter
Author: Master Jones
Price: 500
```

Day 14

In [32]:

```
import numpy as np
class Difference:
    def __init__(self, a):
        self.elements=a
        self.max_diff = 0
    def calc_diff(self):
        self.max_diff = np.max(a) - np.min(a)
        return self.max_diff
a = [int(e) for e in input("Enter numbers: ").split(' ')]
diff = Difference(a)
print(diff.calc_diff())
```

```
Enter numbers: 1 3 4 6 8 11
10
```

Day 15

In [39]:

```
class Node():
    def __init__ (self, data):
        self.data = data
        self.next = None
class Solution():
    def display(self, head):
        n = head
        while n:
            print(n.data, end="")
            n = n.next
    def insert(self,head,data):
        if head is None:
            head = Node(data)
        elif head.next is None:
            head.next = Node(data)
        else:
            self.insert(head.next, data)
            return head
mylist= Solution()
T=int(input())
head=None
for i in range(T):
    data=int(input())
    head=mylist.insert(head,data)
mylist.display(head)
```

```
4
4
3
2
1
4312
```

Day 16

In [40]:

```
n = input("Enter a string: ")
try:
    if type(int(n)):
        print(int(n))
except:
    print("Bad String")
```

```
Enter a string: hello
Bad String
```

Day 17

In [42]:

```
n = int(input("Enter base number: "))
p = int(input("Enter power: "))
if n<0 or p<0:
    raise Exception("n and p should be non negative")
else:
    print(pow(n,p))
```

```
Enter base number: 4
Enter power: 2
16
```

Day 18

In [47]:

```
import sys
from collections import deque

class Solution:
    def __init__(self):
        self.stack = deque()
        self.queue = deque()

    def pushCharacter(self,char):
        self.stack.append(char)

    def popCharacter(self):
        return self.stack.pop()

    def enqueueCharacter(self,char):
        self.queue.append(char)

    def dequeueCharacter(self):
        return self.queue.popleft();

s=input("Enter a string")
obj=Solution()

l=len(s)
for i in range(l):
    obj.pushCharacter(s[i])
    obj.enqueueCharacter(s[i])

isPalindrome=True
for i in range(l // 2):
    if obj.popCharacter()!=obj.dequeueCharacter():
        isPalindrome=False
        break

if isPalindrome:
    print("The word, "+s+", is a palindrome.")
else:
    print("The word, "+s+", is not a palindrome.")
```

```
Enter a stringmadam
The word, madam, is a palindrome.
```

Day 19

In [51]:

```
class AdvancedArithmetic(object):
    def divisorSum(n):
        raise NotImplementedError

class Calculator(AdvancedArithmetic):
    def divisorSum(self, n):
        s = 0
        for i in range(1,n+1):
            if n%i == 0:
                s=s+i
        return s
n = int(input())
my_calculator = Calculator()
s = my_calculator.divisorSum(n)
print("I implemented: " + type(my_calculator).__bases__[0].__name__ )
print(s)
```

```
12
I implemented: AdvancedArithmetic
28
```

Day 20

In [65]:

```
import math
import os
import random
import re
import sys

n = int(input().strip())
a = list(map(int, input().strip().split(' ')))
numberOfSwaps = 0
```

```
for i in range(n):
    for j in range(0, n-1):
        if (a[j] > a[j + 1]):
            temp=a[j]
            a[j] = a[j+1]
            a[j+1] = temp
            numberOfSwaps += 1
    if (numberOfSwaps == 0):
        break
print( "Array is sorted in " + str(numberOfSwaps) + " swaps." )
print( "First Element: " + str(a[0]) )
print( "Last Element: " +  str(a[n-1]) )
```

```
3
3 2 1
Array is sorted in 3 swaps
First Element: 1
Last element: 3
```

Day 21

Unable to do in Python

Day 22

In [72]:

```
class Node:
    def __init__(self,data):
        self.right=self.left=None
        self.data = data
class Solution:
    def insert(self,root,data):
        if root==None:
            return Node(data)
        else:
            if data<=root.data:
                cur=self.insert(root.left,data)
                root.left=cur
            else:
                cur=self.insert(root.right,data)
                root.right=cur
        return root

    def getHeight(self,root):
        if root is None or (root.left is None and root.right is None):
            return 0
        else:
            return max(self.getHeight(root.left),self.getHeight(root.right))+1



T=int(input())
myTree=Solution()
root=None
for i in range(T):
    data=int(input())
    root=myTree.insert(root,data)
height=myTree.getHeight(root)
print(height)
```

```
3
2
1
4

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-72-724943b167c0> in <module>
```

```
     30     data=int(input())
     31     root=myTree.insert(root,data)
---> 32 height=myTree.getHeight(root)
     33 print(height)

<ipython-input-72-724943b167c0> in getHeight(self, root)
     20             return 0
     21         else:
---> 22             return max(self.getHeight(root.left),self.getHeight(root.right))+1
     23
     24

TypeError: 'int' object is not callable
```

Day 23

In [75]:

```python
import sys

class Node:
    def __init__(self,data):
        self.right=self.left=None
        self.data = data

class Solution:
    def insert(self,root,data):
        if root==None:
            return Node(data)
        else:
            if data<=root.data:
                cur=self.insert(root.left,data)
                root.left=cur
            else:
                cur=self.insert(root.right,data)
                root.right=cur
        return root

    def levelOrder(self,root):
        output = ""
        queue = [root]
        while queue:
            current = queue.pop(0)
            output += str(current.data) + " "
            if current.left:
                queue.append(current.left)
                if current.right:
                    queue.append(current.right)
                    print(output[:-1])

T=int(input())
myTree=Solution()
root=None
for i in range(T):
    data=int(input())
    root=myTree.insert(root,data)
myTree.levelOrder(root)
```

```
6
3
5
4
7
2
1
3 2 5 1 4 7
```

Day 24

In [83]:

```
class Node:
    def __init__(self,data):
        self.data = data
        self.next = None
class Solution:
    def insert(self,head,data):
        p = Node(data)
        if head==None:
            head=p
        elif head.next==None:
            head.next=p
        else:
            start=head
            while(start.next!=None):
                start=start.next
                start.next=p
        return  head
    def display(self,head):
        current = head
        while current:
            print(current.data,end=' ')
            current = current.next
    def removeDuplicates(self,head):
        current = head
        while (current.next):
            if (current.data == current.next.data):
                current.next = current.next.next
            else:
                current = current.next
        return head

mylist= Solution()
T=int(input())
head=None
for i in range(T):
    data=int(input())
    head=mylist.insert(head,data)
head=mylist.removeDuplicates(head)
mylist.display(head);
```

```
5
1
2
2
3
4
1 2 3 4
```

Day 25

In [86]:

```
import math

def check_prime(num):
    if num==1:
        return "Not prime"
    for x in range(2, int(math.sqrt(num))+1):
        if num % x ==0:
            return "Not prime"
    return "Prime"
t = int(input("Enter number of values: "))
for i in range(t):
    number = int(input("Enter the values: "))
    print(check_prime(number))
```

```
Enter number of values: 2
Enter the values: 7
Prime
```

```
Enter the values: 4
Not prime
```

Day 26

In [90]:

```
d1, m1, y1 = input().split(' ')
d1 = int(d1)
m1 = int(m1)
y1 = int(y1)
d2, m2, y2 = input().split(' ')
d2 = int(d2)
m2 = int(m2)
y2 = int(y2)
fine = 0
if(y2==y1):
    if(m2 < m1):
        fine = (m1 - m2) * 500
    elif((m2 == m1) and (d2 < d1)):
        fine = (d1 - d2) * 15
elif(y2 < y1):
    fine = 10000
print(fine)
```

```
9 6 2015
6 6 2015
45
```

Day 27

In [93]:

```
def minimum_index(seq):
    if len(seq) == 0:
        raise ValueError("Cannot get the minimum value index from an empty sequence")
    min_idx = 0
    for i in range(1, len(seq)):
        if seq[i] < seq[min_idx]:
            min_idx = i
    return min_idx
print('OK')
class TestDataEmptyArray(object):

    @staticmethod
    def get_array():
        return []

class TestDataUniqueValues(object):

    @staticmethod
    def get_array():
        return [7, 4, 3, 8, 14]

    @staticmethod
    def get_expected_result():
        return 2

class TestDataExactlyTwoDifferentMinimums(object):

    @staticmethod
    def get_array():
        return [7, 4, 3, 8, 3, 14]

    @staticmethod
    def get_expected_result():
        return 2
```

```
OK
```

Day 28

In [97]:

```python
import sys
import re
N = int(input().strip())
names = []
for a0 in range(N):
    firstName,emailID = input().strip().split(' ')
    firstName,emailID = [str(firstName),str(emailID)]
    match = re.search(r'[\w\.-]+@gmail.com', emailID)

    if match:
        names.append(firstName)
names.sort()
for name in names:
    print( name )
```

```
3
calvin calvin16@gmail.com
nmims college@nmims.com
raj rajrao@gmail.com
calvin
raj
```

Day 29

In [99]:

```python
import sys

t = int(input().strip())
for a0 in range(t):
    n, k = input().strip().split(' ')
    n, k = [int(n), int(k)]
    print(k-1 if ((k-1) | k) <= n else k-2)
```

```
3
5 2
1
8 5
4
2 7
5
```