



School of Mechanical and Manufacturing Engineering (SMME)
National University of Sciences and Technology (NUST)

ASSIGNMENT # 3

Submitted by: Aina Zulfiqar

Registration #: 450192

Program: MS – Robotics & Intelligent Machine Engineering

Batch: Fall – 2023

Course: CSE 860 – Artificial Intelligence

Submitted to: Dr. Yasar Ayaz

Dated: January 1, 2024

PYTHON CHALLENGES

1 Medium Challenges

1.1 Write a Function

a. Code:

```
def is_leap(year):
    if 1900 <= year <= 10**5:
        leap = False
        if year % 4 == 0 and year % 100 != 0:
            leap = True
        if year % 400 == 0:
            leap = True
    return leap
```

```
year = int(input())
print(is_leap(year))
```

b. Screenshots:



The screenshot shows a Python IDE with a dark theme. The code is as follows:

```
1  def is_leap(year):
2      if 1900 <= year <= 10**5:
3          leap = False
4          if year % 4 == 0 and year % 100 != 0:
5              leap = True
6          if year % 400 == 0:
7              leap = True
8      return leap
9
10 year = int(input())
11 print(is_leap(year))
```

The IDE interface includes a "Change Theme" button, a "Language" dropdown menu set to "Python 3", and a status bar at the bottom right indicating "Line: 11 Col: 21".

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

Input (stdin)

[Download](#)

| | |
|---|------|
| 1 | 1990 |
|---|------|

Your Output (stdout)

| | |
|---|-------|
| 1 | False |
|---|-------|

Expected Output

[Download](#)

| | |
|---|-------|
| 1 | False |
|---|-------|

1.2 The Minion Game

a. Code:

```
def minion_game(string: str) -> None:
    kevin = stuart = 0
    length: int = len(string)
    for i, char in enumerate(string):
        points: int = length - i
        if char in {"A", "E", "I", "O", "U"}:
            kevin += points
        else:
            stuart += points
    if kevin == stuart:
        print("Draw")
    else:
        print(*("Stuart", stuart) if stuart > kevin else ("Kevin", kevin))

if __name__ == '__main__':
    s = input()
    minion_game(s)
```

b. Screenshots:

```

1  def minion_game(string: str) -> None:
2      kevin = stuart = 0
3      length: int = len(string)
4      for i, char in enumerate(string):
5          points: int = length - i
6          if char in {"A", "E", "I", "O", "U"}:
7              kevin += points
8          else:
9              stuart += points
10     if kevin == stuart:
11         print("Draw")
12     else:
13         print(*("Stuart", stuart) if stuart > kevin else ("Kevin", kevin))
14
15  if __name__ == '__main__':
16      s = input()
17      minion_game(s)

```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

Input (stdin)

[Download](#)

```
1 BANANA
```

Your Output (stdout)

```
1 Stuart 12
```

Expected Output

[Download](#)

```
1 Stuart 12
```

1.3 Merge the Tools!

a. Code:

```

def merge_the_tools(string, k):
    l = len(string)//k
    for i in range(l):
        print("".join(dict.fromkeys(string[i*k:(i*k)+k])))

if __name__ == '__main__':

```

```
string, k = input(), int(input())
merge_the_tools(string, k)
```

b. Screenshots:

```
1 def merge_the_tools(string, k):
2     l = len(string)//k
3     for i in range(l):
4         print(''.join(dict.fromkeys(string[i*k:(i*k)+k])))
5
6 ✓ if __name__ == '__main__':
7     string, k = input(), int(input())
8     merge_the_tools(string, k)
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

Input (stdin)

[Download](#)

```
1 AABCAAADA
2 3
```

Your Output (stdout)

```
1 AB
2 CA
3 AD
```

Expected Output

[Download](#)

```
1 AB
2 CA
3 AD
```

1.4 Time Delta

a. Code:

```
import math
import os
import random
import re
import sys
from datetime import datetime
def time_delta(t1, t2):
    format_ = '%a %d %b %Y %H:%M:%S %z'
    t1 = datetime.strptime(t1, format_)
    t2 = datetime.strptime(t2, format_)
    return str(int(abs((t1-t2).total_seconds())))
if __name__ == '__main__':
    fptr = open(os.environ['OUTPUT_PATH'], 'w')

    t = int(input())

    for t_itr in range(t):
        t1 = input()

        t2 = input()

        delta = time_delta(t1, t2)

        fptr.write(delta + '\n')

    fptr.close()
```

b. Screenshots:

```

2  import math
3  import os
4  import random
5  import re
6  import sys
7  from datetime import datetime
8  def time_delta(t1, t2):
9      format_ = '%a %d %b %Y %H:%M:%S %z'
10     t1 = datetime.strptime(t1, format_)
11     t2 = datetime.strptime(t2, format_)
12     return str(int(abs((t1-t2).total_seconds())))
13 if __name__ == '__main__':
14     fptr = open(os.environ['OUTPUT_PATH'], 'w')
15
16     t = int(input())
17
18     for t_itr in range(t):
19         t1 = input()
20
21         t2 = input()
22
23         delta = time_delta(t1, t2)
24
25         fptr.write(delta + '\n')
26
27     fptr.close()

```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

| | |
|---|--------------------------------|
| 1 | 2 |
| 2 | Sun 10 May 2015 13:54:36 -0700 |
| 3 | Sun 10 May 2015 13:54:36 -0000 |
| 4 | Sat 02 May 2015 19:54:36 +0530 |
| 5 | Fri 01 May 2015 13:54:36 -0000 |

Your Output (stdout)

| | |
|---|-------|
| 1 | 25200 |
| 2 | 88200 |

Expected Output

| | |
|---|-------|
| 1 | 25200 |
| 2 | 88200 |

[Download](#)

1.5 Find Angle MBC

a. Code:

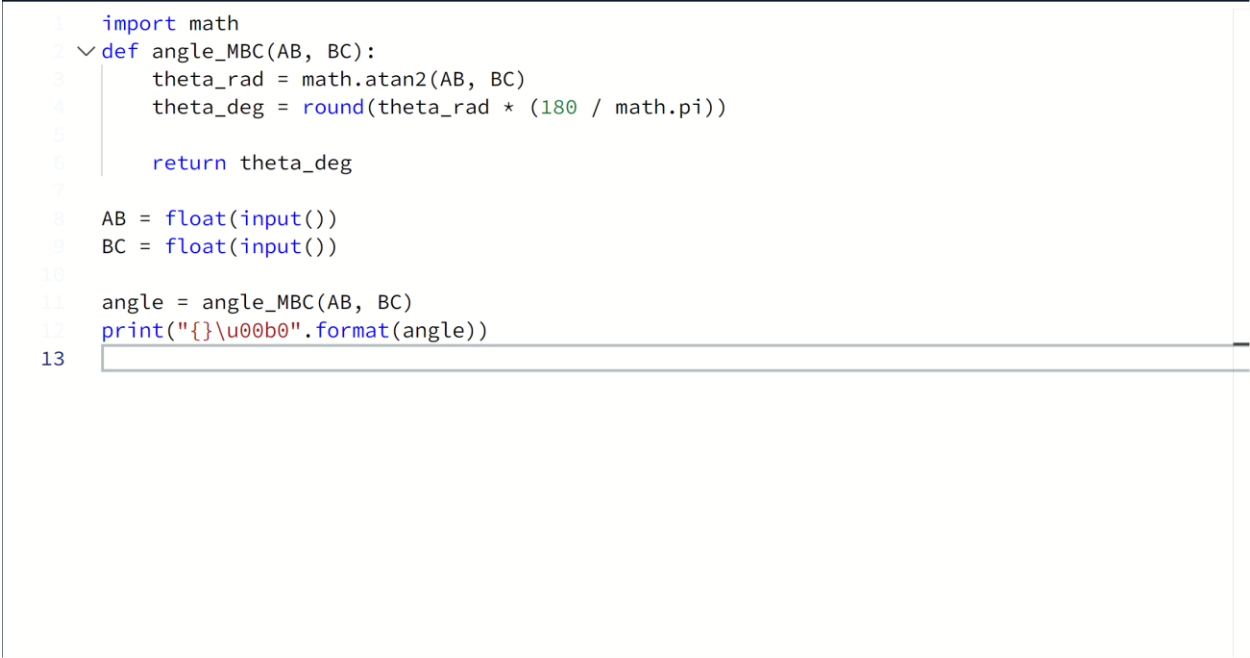
```
import math
def angle_MBC(AB, BC):
    theta_rad = math.atan2(AB, BC)
    theta_deg = round(theta_rad * (180 / math.pi))

    return theta_deg

AB = float(input())
BC = float(input())

angle = angle_MBC(AB, BC)
print("{}\u00b0".format(angle))
```

b. Screenshots:



```
1 import math
2 def angle_MBC(AB, BC):
3     theta_rad = math.atan2(AB, BC)
4     theta_deg = round(theta_rad * (180 / math.pi))
5
6     return theta_deg
7
8 AB = float(input())
9 BC = float(input())
10
11 angle = angle_MBC(AB, BC)
12 print("{}\u00b0".format(angle))
13
```


Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

Input (stdin)

[Download](#)

```
1 10
2 10
```

Your Output (stdout)

```
1 45°
```

Expected Output

[Download](#)

```
1 45°
```

1.6 No Idea!

a. Code:

```
if __name__ == "__main__":
    happy = 0
    n, m = map(int, input().strip().split(' '))
    elements_arr = list(map(int, input().strip().split(' ')))

    A = set(map(int, input().strip().split(' ')))
    B = set(map(int, input().strip().split(' ')))

    for i in elements_arr:
        if i in A:
            happy += 1
        elif i in B:
            happy -= 1
    print(happy)
```

b. Screenshots:

```

1  ∨ if __name__ == "__main__":
2      happy = 0
3      n, m = map(int, input().strip().split(' '))
4      elements_arr = list(map(int, input().strip().split(' ')))
5
6      A = set(map(int, input().strip().split(' ')))
7      B = set(map(int, input().strip().split(' ')))
8
9  ∨   for i in elements_arr:
10 ∨       if i in A:
11         happy += 1
12 ∨       elif i in B:
13         happy -= 1
14     print(happy)
15

```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✔ **Sample Test case 0**

Input (stdin)

Download

| | |
|---|-------|
| 1 | 3 2 |
| 2 | 1 5 3 |
| 3 | 3 1 |
| 4 | 5 7 |

Your Output (stdout)

Download

| | |
|---|---|
| 1 | 1 |
|---|---|

Expected Output

Download

| | |
|---|---|
| 1 | 1 |
|---|---|

1.7 Word Order

a. Code:

```

n=int(input())
words=[input() for i in range(n)]
occur={}

```

```
for word in words:
    occur[word]=0
for word in words:
    occur[word]+=1
print(len(occur))
occurences=occur.values()
for i in occurences:
    print(i,end=" ")
```

b. Screenshots:

```
1  n=int(input())
2  words=[input() for i in range(n)]
3  occur={}
4  √ for word in words:
5  |     occur[word]=0
6  √ for word in words:
7  |     occur[word]+=1
8  print(len(occur))
9  occurences=occur.values()
10 √ for i in occurences:
11 |     print(i,end=" ")
12
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

| | |
|---|---------|
| 1 | 4 |
| 2 | bcdef |
| 3 | abcdefg |
| 4 | bcde |
| 5 | bcdef |

Your Output (stdout)

| | |
|---|-------|
| 1 | 3 |
| 2 | 2 1 1 |

Expected Output

| | |
|---|-------|
| 1 | 3 |
| 2 | 2 1 1 |

[Download](#)

1.8 Compress the String!

a. Code:

```
from itertools import groupby
for a, b in groupby(input()):
    print("(%d, %d)" % (len(list(b)), int(a)), end=' ')
```

b. Screenshots:

```

1 from itertools import groupby
2 ✓ for a, b in groupby(input()):
3     print("(%d, %d)" % (len(list(b)), int(a)), end=' ')
4

```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

Input (stdin)

[Download](#)

1 1222311

Your Output (stdout)

1 (1, 1) (3, 2) (1, 3) (2, 1)

Expected Output

[Download](#)

1 (1, 1) (3, 2) (1, 3) (2, 1)

1.9 Company Logo

a. Code:

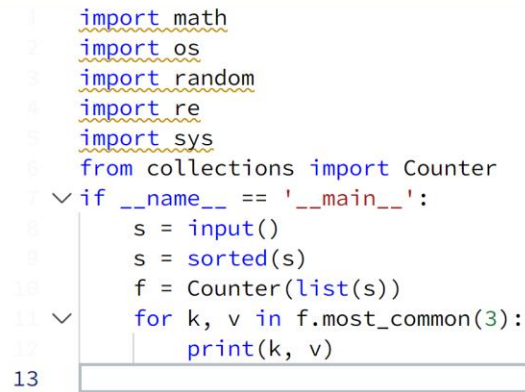
```

import math
import os
import random
import re
import sys
from collections import Counter

```

```
if __name__ == '__main__':  
    s = input()  
    s = sorted(s)  
    f = Counter(list(s))  
    for k, v in f.most_common(3):  
        print(k, v)
```

b. Screenshots:



```
1  import math  
2  import os  
3  import random  
4  import re  
5  import sys  
6  from collections import Counter  
7  if __name__ == '__main__':  
8      s = input()  
9      s = sorted(s)  
10     f = Counter(list(s))  
11     for k, v in f.most_common(3):  
12         print(k, v)
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

Input (stdin)

[Download](#)

```
1 aabbccde
```

Your Output (stdout)

```
1 b 3
2 a 2
3 c 2
```

Expected Output

[Download](#)

```
1 b 3
2 a 2
3 c 2
```

1.10 Piling Up!

a. Code:

```
t = int(input())

for _ in range(t):
    num, cubes = int(input()), list(map(int,input().split()))
    answer = "Yes"

    while len(cubes) > 1:
        if cubes[0] >= cubes[-1]:
            large = cubes[0]
            cubes.pop(0)
        else:
            large = cubes[-1]
            cubes.pop(-1)
        if large < cubes[0] or large < cubes[-1]:
            answer = 'No'
            break

    print(answer)
```

b. Screenshots:

```

1  t = int(input())
2
3  ✓ for _ in range(t):
4      num, cubes = int(input()), list(map(int, input().split()))
5      answer = "Yes"
6
7  ✓ while len(cubes) > 1:
8      ✓ if cubes[0] >= cubes[-1]:
9          large = cubes[0]
10         cubes.pop(0)
11     ✓ else:
12         large = cubes[-1]
13         cubes.pop(-1)
14     ✓ if large < cubes[0] or large < cubes[-1]:
15         answer = 'No'
16         break
17
18     print(answer)
19

```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

```

1  2
2  6
3  4 3 2 1 3 4
4  3
5  1 3 2

```

Your Output (stdout)

```

1  Yes
2  No

```

Expected Output

```

1  Yes
2  No

```

[Download](#)

1.11 Triangle Quest 2

a. Code:

```
for i in range(1,int(input())+1):
```



```
print (((10 ** i - 1) // 9) ** 2)
```

b. Screenshots:

The screenshot shows a code editor with the following Python code:

```
1  ✓ for i in range(1, int(input())+1):  
2  print (((10 ** i - 1) // 9) ** 2)
```

Below the code editor is a dark-themed panel with the following sections:

- Congratulations!**
You have passed the sample test cases. Click the submit button to run your code against all the test cases.
- Sample Test case 0** (with a green checkmark icon)
- Compiler Message**
Success
- Input (stdin)**
1 5
A [Download](#) link is visible to the right.
- Your Output (stdout)**
1 1
2 121
3 12321
4 1234321
5 123454321

1.12 Iterables and Iterators

a. Code:

```
from itertools import combinations, groupby
```

```
count, letters, select = int(input()), input().split(), int(input())
letters = sorted(letters)
combine = list(combinations(letters, select))
contain = len([c for c in combine if 'a' in c])
print(contain / len(combine))
```

b. Screenshots:

```
1  from itertools import combinations, groupby
2
3  count, letters, select = int(input()), input().split(), int(input())
4  letters = sorted(letters)
5  combine = list(combinations(letters, select))
6  contain = len([c for c in combine if 'a' in c])
7  print(contain / len(combine))
8
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

Input (stdin)

[Download](#)

```
1 4
2 a a c d
3 2
```

Your Output (stdout)

```
1 0.8333333333333334
```

Expected Output

[Download](#)


```
1 0.833333333333
```

1.13 Triangle Quest

a. Code:

```
for i in range(1,int(input())):  
    print((10 ** i-1) // 9 * i)
```

b. Screenshots:



The screenshot shows a code editor with two lines of Python code. Line 1 is a for loop: `for i in range(1,int(input())):`. Line 2 is an indented print statement: `print((10 ** i-1) // 9 * i)`. The code is color-coded: `for` is blue, `i` is green, `in` is blue, `range` is blue, `1` is green, `,` is green, `int` is blue, `(input())` is green, `:` is blue, `print` is blue, `((10 ** i-1) // 9 * i)` is green. A vertical scrollbar is visible on the right side of the editor.

```
1  ∨ for i in range(1,int(input())):  
2  print((10 ** i-1) // 9 * i)
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

Compiler Message

Success

Input (stdin)

[Download](#)

1 5

Your Output (stdout)

1 1
2 22
3 333
4 4444

1.14 Classes: Dealing with Complex Numbers

a. Code:

```
import math
class Complex(object):
    def __init__(self, real, imaginary):
        self.real = real
        self.imaginary = imaginary
    def __add__(self, no):
        return Complex(self.real + no.real , self.imaginary + no.imaginary)
    def __sub__(self, no):
        return Complex(self.real - no.real , self.imaginary - no.imaginary)
    def __mul__(self, no):
        prod = complex(self.real , self.imaginary)*complex(no.real , no.imaginary)
        return Complex(prod.real , prod.imag)
    def __truediv__(self, no):
        div = complex(self.real , self.imaginary)/complex(no.real , no.imaginary)
        return Complex(div.real , div.imag)
    def mod(self):
        m = math.sqrt(self.real**2 + self.imaginary**2)
        return Complex(m,0)
    def __str__(self):
        if self.imaginary == 0:
```

```

        result = "%.2f+0.00i" % (self.real)
    elif self.real == 0:
        if self.imaginary >= 0:
            result = "0.00+%.2fi" % (self.imaginary)
        else:
            result = "0.00-%.2fi" % (abs(self.imaginary))
    elif self.imaginary > 0:
        result = "%.2f+%.2fi" % (self.real, self.imaginary)
    else:
        result = "%.2f-%.2fi" % (self.real, abs(self.imaginary))
    return result

if __name__ == '__main__':
    c = map(float, input().split())
    d = map(float, input().split())
    x = Complex(*c)
    y = Complex(*d)
    print(*map(str, [x+y, x-y, x*y, x/y, x.mod(), y.mod()]), sep='\n')

```

b. Screenshots:



```

1  import math
2  class Complex(object):
3      def __init__(self, real, imaginary):
4          self.real = real
5          self.imaginary = imaginary
6      def __add__(self, no):
7          return Complex(self.real + no.real, self.imaginary + no.imaginary)
8      def __sub__(self, no):
9          return Complex(self.real - no.real, self.imaginary - no.imaginary)
10     def __mul__(self, no):
11         prod = complex(self.real, self.imaginary)*complex(no.real, no.
12         imaginary)
13         return Complex(prod.real, prod.imag)
14     def __truediv__(self, no):
15         div = complex(self.real, self.imaginary)/complex(no.real, no.imaginary)
16         return Complex(div.real, div.imag)
17     def mod(self):
18         m = math.sqrt(self.real**2 + self.imaginary**2)
19         return Complex(m,0)

```

```

19     def __str__(self):
20         if self.imaginary == 0:
21             result = "%.2f+0.00i" % (self.real)
22         elif self.real == 0:
23             if self.imaginary >= 0:
24                 result = "0.00+%.2fi" % (self.imaginary)
25             else:
26                 result = "0.00-%.2fi" % (abs(self.imaginary))
27         elif self.imaginary > 0:
28             result = "%.2f+%.2fi" % (self.real, self.imaginary)
29         else:
30             result = "%.2f-%.2fi" % (self.real, abs(self.imaginary))
31         return result
32
33     if __name__ == '__main__':
34         c = map(float, input().split())
35         d = map(float, input().split())
36         x = Complex(*c)
37         y = Complex(*d)
38         print(*map(str, [x+y, x-y, x*y, x/y, x.mod(), y.mod()]), sep='\n')

```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

Input (stdin)

[Download](#)

✓ Sample Test case 1

Your Output (stdout)

```

1  7.00+7.00i
2  -3.00-5.00i
3  4.00+17.00i
4  0.26-0.11i
5  2.24+0.00i
6  7.81+0.00i

```

1.15 Athlete Sort

a. Code:

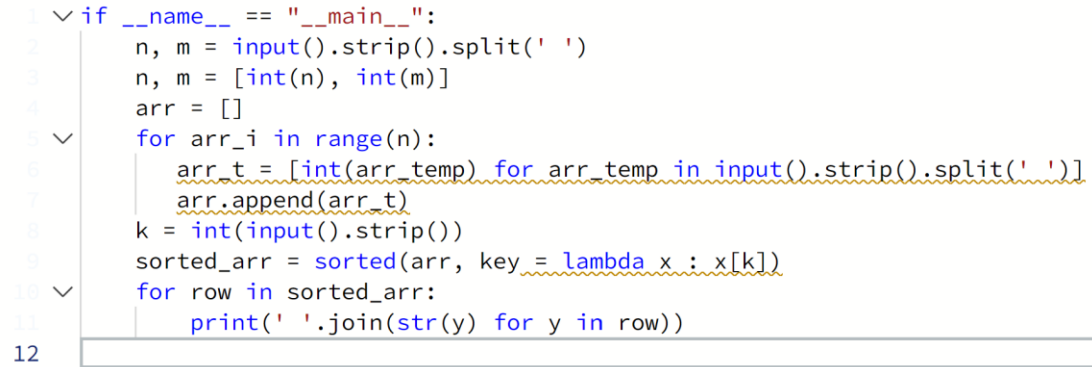
```

if __name__ == "__main__":
    n, m = input().strip().split(' ')
    n, m = [int(n), int(m)]
    arr = []
    for arr_i in range(n):
        arr_t = [int(arr_temp) for arr_temp in input().strip().split(' ')]

```

```
arr.append(arr_t)
k = int(input().strip())
sorted_arr = sorted(arr, key = lambda x : x[k])
for row in sorted_arr:
    print(' '.join(str(y) for y in row))
```

b. Screenshots:



```
1  ✓ if __name__ == "__main__":
2      n, m = input().strip().split(' ')
3      n, m = [int(n), int(m)]
4      arr = []
5  ✓  for arr_i in range(n):
6      |     arr_t = [int(arr_temp) for arr_temp in input().strip().split(' ')]
7      |     arr.append(arr_t)
8      k = int(input().strip())
9      sorted_arr = sorted(arr, key = lambda x : x[k])
10 ✓  for row in sorted_arr:
11      |     print(' '.join(str(y) for y in row))
12
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

```
1 5 3
2 10 2 5
3 7 1 0
4 9 9 9
5 1 23 12
6 6 5 9
7 1
```

Your Output (stdout)

```
1 7 1 0
2 10 2 5
3 6 5 9
4 9 9 9
5 1 23 12
```

1.16 ginortS

a. Code:

```
print(*sorted(input(), key=lambda c: (c.isdigit() - c.islower(), c in '02468', c)), sep="")
```

b. Screenshots:


```
1 print(*sorted(input(), key=lambda c: (c.isdigit() - c.islower(), c in '02468', c)
2 ), sep='')
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

Input (stdin)

[Download](#)

```
1 Sorting1234
```

Your Output (stdout)

```
1 ginortS1324
```

Expected Output

[Download](#)

```
1 ginortS1324
```

1.17 Validating Email Addresses With a Filter

a. Code:

```
import re
def fun(s):
    pattern = re.compile("^[\\w-]+@[0-9a-zA-Z]+\\.[a-z]{1,3}$")
    return pattern.match(s)
def filter_mail(emails):
```

```

    return list(filter(fun, emails))

if __name__ == '__main__':
    n = int(input())
    emails = []
    for _ in range(n):
        emails.append(input())

filtered_emails = filter_mail(emails)
filtered_emails.sort()
print(filtered_emails)

```

b. Screenshots:



```

1  import re
2  def fun(s):
3      pattern = re.compile("^[\w-]+@[0-9a-zA-Z]+\.[a-z]{1,3}$")
4      return pattern.match(s)
5  def filter_mail(emails):
6      return list(filter(fun, emails))
7
8  if __name__ == '__main__':
9      n = int(input())
10     emails = []
11     for _ in range(n):
12         emails.append(input())
13
14     filtered_emails = filter_mail(emails)
15     filtered_emails.sort()
16     print(filtered_emails)

```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

✓ Sample Test case 1

Input (stdin)

[Download](#)

```
1 3
2 lara@hackerrank.com
3 brian-23@hackerrank.com
4 britts_54@hackerrank.com
```

Your Output (stdout)

```
1 ['brian-23@hackerrank.com', 'britts_54@hackerrank.com',
  'lara@hackerrank.com']
```

Expected Output

[Download](#)

```
1 ['brian-23@hackerrank.com', 'britts_54@hackerrank.com',
  'lara@hackerrank.com']
```

1.18 Reduce Function

a. Code:

```
from fractions import Fraction
from functools import reduce
```

```
def product(fracs):
    t = Fraction(reduce(lambda x, y: x * y, fracs))
    return t.numerator, t.denominator
```

```
if __name__ == '__main__':
    fracs = []
    for _ in range(int(input())):
        fracs.append(Fraction(*map(int, input().split())))
    result = product(fracs)
    print(*result)
```

b. Screenshots:

```

1  ✓ from fractions import Fraction
2    from functools import reduce
3
4    def product(fracs):
5        t = Fraction(reduce(lambda x, y: x * y, fracs))
6        return t.numerator, t.denominator
7
8  ✓ if __name__ == '__main__':
9      fracs = []
10     for _ in range(int(input())):
11         fracs.append(Fraction(*map(int, input().split())))
12     result = product(fracs)
13     print(*result)

```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

Input (stdin)

[Download](#)

```

1  3
2  1 2
3  3 4
4  10 6

```

Your Output (stdout)

```

1  5 8

```

Expected Output

[Download](#)

```

1  5 8

```

1.19 Regex Substitution

a. Code:

```
import re
```

```
for i in range(int(input())):
```

```
s = re.sub("(?<=\\s)&&(?=\\s)", "and", input())
print(re.sub("(?<=\\s)\\|\\|(?=\\s)", "or", s))
```

b. Screenshots:

```
1 import re
2
3 ✓ for i in range(int(input())):
4     s = re.sub("(?<=\\s)&&(?=\\s)", "and", input())
5     print(re.sub("(?<=\\s)\\|\\|(?=\\s)", "or", s))
6
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ **Sample Test case 0**

Your Output (stdout)

```
1 a = 1;
2 b = input();
3
4 if a + b > 0 and a - b < 0:
5     start()
6 elif a*b > 10 or a/b < 1:
7     stop()
8 print set(list(a)) | set(list(b))
9 #Note do not change &&& or ||| or & or |
10 #Only change those '&&' which have space on both sides.
11 #Only change those '||' which have space on both sides.
```

1.20 Validating Credit Card Numbers

a. Code:

```
import re
n = int(input())
for t in range(n):
    credit = input().strip()
    credit_removed_hiphen = credit.replace('-', '')

    valid = True

    length_16 = bool(re.match(r'^[4-6]\d{15}$', credit))
    length_19 = bool(re.match(r'^[4-6]\d{3}-\d{4}-\d{4}-\d{4}$', credit))
    consecutive = bool(re.findall(r'(?=(\d)\1\1\1)', credit_removed_hiphen))

    if length_16 == True or length_19 == True:
        if consecutive == True:
            valid = False
        else:
            valid = False
    if valid == True:
        print('Valid')
    else:
        print('Invalid')
```

b. Screenshots:

```

1  import re
2  n = int(input())
3  for t in range(n):
4      credit = input().strip()
5      credit_removed_hiphen = credit.replace('-', '')
6
7      valid = True
8
9      length_16 = bool(re.match(r'^[4-6]\d{15}$', credit))
10     length_19 = bool(re.match(r'^[4-6]\d{3}-\d{4}-\d{4}-\d{4}$', credit))
11     consecutive = bool(re.findall(r'(?=(\d)\1\1\1)', credit_removed_hiphen))
12
13     if length_16 == True or length_19 == True:
14         if consecutive == True:
15             valid = False
16     else:
17         valid = False
18     if valid == True:
19         print('Valid')
20     else:
21         print('Invalid')
22

```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

Input (stdin)

[Download](#)

```

1  6
2  4123456789123456
3  5123-4567-8912-3456
4  61234-567-8912-3456
5  4123356789123456
6  5133-3367-8912-3456
7  5123 - 3567 - 8912 - 3456

```

Sample Test case 0

Your Output (stdout)

| | |
|---|----------------|
| 1 | Valid |
| 2 | Valid |
| 3 | Invalid |
| 4 | Valid |
| 5 | Invalid |
| 6 | Invalid |

Expected Output

| | |
|---|----------------|
| 1 | Valid |
| 2 | Valid |
| 3 | Invalid |
| 4 | Valid |

[Download](#)

1.21 Words Score

a. Code:

```
def is_vowel(letter):
    return letter in ['a', 'e', 'i', 'o', 'u', 'y']
```

```
def score_words(words):
    score = 0
    for word in words:
        num_vowels = 0
        for letter in word:
            if is_vowel(letter):
                num_vowels += 1
        if num_vowels % 2 == 0:
            score += 2
        else:
            score += 1
    return score
```

```
n = int(input())
words = input().split()
print(score_words(words))
```

b. Screenshots:


```

1  def is_vowel(letter):
2      return letter in ['a', 'e', 'i', 'o', 'u', 'y']
3
4  def score_words(words):
5      score = 0
6      for word in words:
7          num_vowels = 0
8          for letter in word:
9              if is_vowel(letter):
10                 num_vowels += 1
11             if num_vowels % 2 == 0:
12                 score += 2
13             else:
14                 score += 1
15         return score
16
17  n = int(input())
18  words = input().split()
19  print(score_words(words))
20

```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

✓ Sample Test case 1

Input (stdin)

[Download](#)

```

1  2
2  hacker book

```

Your Output (stdout)

```

1  4

```

Expected Output

[Download](#)

```

1  4

```

1.22 Default Arguments

a. Code:

```

class EvenStream(object):
    def __init__(self):
        self.current = 0

    def get_next(self):

```

```

        to_return = self.current
        self.current += 2
        return to_return

class OddStream(object):
    def __init__(self):
        self.current = 1

    def get_next(self):
        to_return = self.current
        self.current += 2
        return to_return

def print_from_stream(n, stream=None):
    if stream is None:
        stream = EvenStream()

    for _ in range(n):
        print(stream.get_next())
raw_input = input

queries = int(input())
for _ in range(queries):
    stream_name, n = input().split()
    n = int(n)
    if stream_name == "even":
        print_from_stream(n)
    else:
        print_from_stream(n, OddStream())

```

b. Screenshots:

```

1  class EvenStream(object):
2      def __init__(self):
3          self.current = 0
4
5      def get_next(self):
6          to_return = self.current
7          self.current += 2
8          return to_return
9
10 class OddStream(object):
11     def __init__(self):
12         self.current = 1
13
14     def get_next(self):
15         to_return = self.current
16         self.current += 2
17         return to_return
18
19
20 def print_from_stream(n, stream=None):
21     if stream is None:
22         stream = EvenStream()
23
24     for _ in range(n):
25         print(stream.get_next())
26 raw_input = input
27
28
29 queries = int(input())
30 for _ in range(queries):
31     stream_name, n = input().split()
32     n = int(n)
33     if stream_name == "even":
34         print_from_stream(n)
35     else:
36         print_from_stream(n, OddStream())
37

```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

4 odd 5

Your Output (stdout)

| | |
|----|---|
| 1 | 1 |
| 2 | 3 |
| 3 | 0 |
| 4 | 2 |
| 5 | 4 |
| 6 | 1 |
| 7 | 3 |
| 8 | 5 |
| 9 | 7 |
| 10 | 9 |

2 Hard Challenges

2.1 Maximize It!

a. Code:

```
import itertools
line = input()
K = int(line.split()[0])
M = int(line.split()[1])
N = []
for i in range(K):
    l = input().split()
    l = [ int(n) for n in l ]
    l = l[1:]
    N.append(l)
pro = list( itertools.product( *N ) )
maxi = 0
for item in pro:
    sum=0
    for num in item:
        sum += num**2
    modu = sum % M
```

```
if(modu > maxi):
    maxi = modu
print(maxi)
```

b. Screenshots:

```
1 import itertools
2 line = input()
3 K = int(line.split()[0])
4 M = int(line.split()[1])
5 N = []
6 for i in range(K):
7     l = input().split()
8     l = [int(n) for n in l]
9     l = l[1:]
10    N.append(l)
11    pro = list(itertools.product(*N))
12    maxi = 0
13    for item in pro:
14        sum=0
15        for num in item:
16            sum += num**2
17        modu = sum % M
18        if(modu > maxi):
19            maxi = modu
20    print(maxi)
21
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

Input (stdin)

[Download](#)

```
1 3 1000
2 2 5 4
3 3 7 8 9
4 5 5 7 8 9 10
```

Your Output (stdout)

```
1 206
```

Expected Output

[Download](#)

```
1 206
```

2.2 Validating Postal Codes

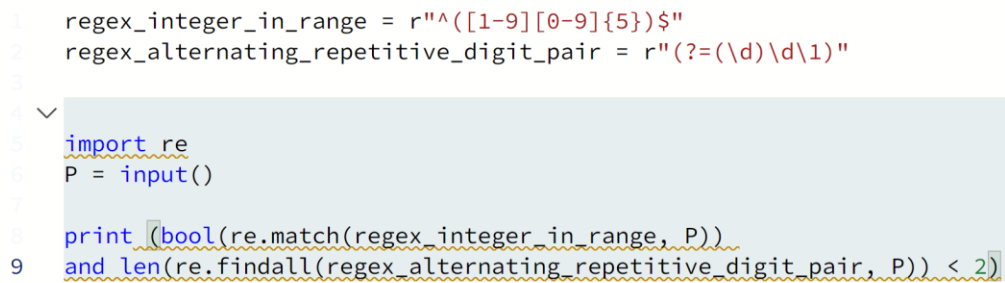
a. Code:

```
regex_integer_in_range = r"^([1-9][0-9]{5})$"
regex_alternating_repetitive_digit_pair = r"(?=(\d)\d\1)"

import re
P = input()

print (bool(re.match(regex_integer_in_range, P))
and len(re.findall(regex_alternating_repetitive_digit_pair, P)) < 2)
```

b. Screenshots:



```
1 regex_integer_in_range = r"^([1-9][0-9]{5})$"
2 regex_alternating_repetitive_digit_pair = r"(?=(\d)\d\1)"
3
4 ✓
5 import re
6 P = input()
7
8 print (bool(re.match(regex_integer_in_range, P))
9 and len(re.findall(regex_alternating_repetitive_digit_pair, P)) < 2)
```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

Input (stdin)

[Download](#)

| | |
|---|--------|
| 1 | 110000 |
|---|--------|

Your Output (stdout)

| | |
|---|-------|
| 1 | False |
|---|-------|

Expected Output

[Download](#)

| | |
|---|-------|
| 1 | False |
|---|-------|

2.3 Matrix Script

a. Code:

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

first_multiple_input = input().rstrip().split()
n = int(first_multiple_input[0])
m = int(first_multiple_input[1])
matrix = []
t = []
for _ in range(n):
    matrix_item = [x for x in input()]
    matrix.append(matrix_item)

for i in range(m):
    for j in range(n):
        t.append(matrix[j][i])
s = ".join(t)
path = re.compile(r'\b[ !@#$%^&]+\b', re.M)
k = re.sub(path, ' ', s)
print(k)
```

b. Screenshots:

```

1  import math
2  import os
3  import random
4  import re
5  import sys
6
7  first_multiple_input = input().rstrip().split()
8  n = int(first_multiple_input[0])
9  m = int(first_multiple_input[1])
10 matrix = []
11 t = []
12 for _ in range(n):
13     matrix_item = [x for x in input()]
14     matrix.append(matrix_item)
15
16 for i in range(m):
17     for j in range(n):
18         t.append(matrix[j][i])
19 s = ''.join(t)
20 path = re.compile(r'\b[ !@#$%&]+\b', re.M)
21 k = re.sub(path, ' ', s)
22 print(k)
23

```

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

```

2  Tsi
3  h%x
4  i #
5  sM
6  $a
7  #t%
8  ir!

```

Your Output (stdout)

```

1  This is Matrix#  %!

```

Expected Output

```

1  This is Matrix#  %!

```

Download

GitHub Repository Link: https://github.com/AinaZed/AinaZulfiqar_450192_AI_Assignment3/