1806554 Assignment (Python)

In [1]:

```python
print("Hola Mundo")
```

```
Hola Mundo
```

In [2]:

```python
r = int(input())
w = int(input())
h = int(input())
aoc = 3.14*r*r
poc = 2*3.14*r
aos = h*h
aor = w*h
print(aoc , " " ,poc , " " , aos , " " , aor)
```

```
4
3
2
50.24   25.12   4   6
```

In [3]:

```python
c = int(input())
f = (c * 1.8) + 32
print(f)
```

```
104
219.20000000000002
```

In [4]:

```python
i = int(input())
j = int(input())
k = int(input())
if i>k and i>j :
    print("{} is greatest".format(i))
elif j>i and j>k:
    print("{} is greatest".format(j))
else :
    print("{} is greatest".format(k))
```

```
2
3
4
4 is greatest
```

In [3]:

```python
import cmath
a = int(input())
b = int(input())
c = int(input())
d = (b**2) - (4*a*c)
s1= (-b-cmath.sqrt(d))/(2*a)
s2 = (-b+cmath.sqrt(d))/(2*a)

print("{} and {}".format(s1,s2))
```

```
1
2
3
(-1-1.4142135623730951j) and (-1+1.4142135623730951j)
```

In [6]:

```
for i in range(65,127):
    print(i, " ",chr(i))
```

```
65   A
66   B
67   C
68   D
69   E
70   F
71   G
72   H
73   I
74   J
75   K
76   L
77   M
78   N
79   O
80   P
81   Q
82   R
83   S
84   T
85   U
86   V
87   W
88   X
89   Y
90   Z
91   [
92   \
93   ]
94   ^
95   _
96   `
97   a
98   b
99   c
100   d
101   e
102   f
103   g
104   h
105   i
106   j
107   k
108   l
109   m
110   n
111   o
112   p
113   q
114   r
115   s
116   t
117   u
118   v
```

```
119   w
120   x
121   y
122   z
123   {
124   |
125   }
126   ~
```

In [7]:

```
x = int(input())
fact = 1
for i in range(1,x+1):
    fact *= i
print(fact)
```

```
5
120
```

In [8]:

```
x = int(input())
y = int(input())
while(y>0):
    x,y = y,x%y
print(x)
```

```
14
7
7
```

In [9]:

```
a = int(input())
b = int(input())
print(a**b)
```

```
4
6
4096
```

In [10]:

```
num = int(input())
if num > 1:
    for i in range(2, num):
        if (num % i) == 0:
            print(num, "is not a prime")
            break
    else:
        print(num, "is a prime")
else:
    print(num, "is not a prime")
```

```
4
4 is not a prime
```

In [12]:

```python
num = int(input())
s = 0
t = num
while t > 0:
    d = t % 10
    s += d ** 3
    t //= 10
if num == s:
    print(num,"is an Armstrong number")
else:
    print(num,"is not an Armstrong number")
```

```
158
158 is not an Armstrong number
```

In [13]:

```python
for num in range(1, 1000 + 1):
    s = 0
    t = num
    while t > 0:
        d = t % 10
        s += d ** 3
        t //= 10

    if num == s:
        print(num)
```

```
1
153
370
371
407
```

In [14]:

```python
def nas(num):
    return num == sum([int(x) ** len(str(num)) for x in str(num)])
num = int(input())
nas(num)
```

```
134
```

Out[14]:

```
False
```

In [15]:

```python
def fibo(n):
    a = 0
    b = 1
    c = a+b
    print(0)
    print(1)
    while c<n:
        print(c)
        a = b
        b = c
        c = a+b

n = int(input())
fibo(n)
```

```
10
0
1
1
2
3
5
8
```

In [16]:

```python
def fibo(n):
    a = 0
    b = 1
    c = a+b
    if n == 1:
        return 0
    if n == 2:
        return 1
    else:
        while c<n :
            a = b
            b = c
            c = a+b
    return c
n = int(input())
fibo(n)
```

```
13
```

Out[16]:

```
13
```

In [ ]:

```
# def fact(n):
#     fact = 1
#     for i in range(1,n+1):
#         fact *= i
#     return fact

# def strng(n):
#     s = n
#     ss = 0
#     for i in range(len(str(n))):
#         ss += fact(int(i))
#         print(i)
#    # return s == sum([fact(int(i)) for i in range(len(str(n)))])
# n = int(input())
# strng(n)
```

In [17]:

```
sum1=0
num=int(input())
t=num
while(num):
    i=1
    f=1
    r=num%10
    while(i<=r):
        f=f*i
        i=i+1
    sum1=sum1+f
    num=num//10
if(sum1==t):
    print("strong number")
else:
    print("not strong number")
```

```
145
strong number
```

In [18]:

```
# Without Loop
n = int(input())
print(str(n)[::-1])

# with Loop
rev = 0
while(n>0):
    a = n % 10
    rev = rev * 10 + a
    n = n // 10
print(rev)
```

```
123
321
```

In [19]:

```python
# Without Loop
n = int(input())
r = int(str(n)[::-1])
if n == r:
    print(f'{n} is a palindrome')
else:
    print(f'{n} is not pal')

# With Loop
rev = 0
while(n>0):
    a = n % 10
    rev = rev * 10 + a
    n = n // 10
if n == rev:
    print(f'{n} is a palindrome')
else:
    print(f'{n} is not pal')
```

```
121
121 is a palindrome
```

In [7]:

```python
x = int(input())
print(bin(x).replace("0b",""))
fi = ""
while x != 0:
    rem = x % 2
    x = x // 2
    fi = str(rem) + fi
print("The binary representation is", fi)
print(hex(int(fi))[2:])
def binaryToDecimal(binary):
    binary1 = binary
    decimal, i, n = 0, 0, 0
    while(binary != 0):
        dec = binary % 10
        decimal = decimal + dec * pow(2, i)
        binary = binary//10
        i += 1
    print(decimal)
binaryToDecimal(int(fi))

```

```
12
1100
The binary representation is 1100
44c
12
```

Python Array Questions

In [13]:

```python
def cre(n,c):
    l = []
    for i in range(1,n):
        x = int(input())
        l.append(x)
    print(f'{c} count : ',l.count(c))
    print(l)
n = int(input())
c = int(input())
cre(n,c)
```

```
5
3
1
3
3
4
3 count :  2
[1, 3, 3, 4]
```

In [11]:

```python
def fre(n):
    d = {}
    l = []
    for i in range(1,n):
        x = int(input())
        l.append(x)

    for item in l:
        if item in d:
            d[item] += 1
        else:
            d[item] = 1

    for i,j in d.items():
        print(f'{i} : {j}')
n = int(input())
fre(n)
```

```
9
1
1
1
1
2
2
2
3
1 : 4
2 : 3
3 : 1
```

In [15]:

```python
def chk(n,c):
    l = []
    for i in range(1,n):
        x = int(input())
        l.append(x)
    check = False
    for i in l:
        if i == c:
            print(f'{i} found')
            check = True
            break
    if check == False:
        print(f'not found {c}')
#     for item in l:
#         if item in d:
#             d[item] += 1
#         else:
#             d[item] = 1

#     for i,j in d.items():
#         print(f'{i} : {j}')
n = int(input())
c = int(input())
chk(n,c)
```

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

In [16]:

```python
def splt(n,c):
    l = []
    for i in range(1,n):
        x = int(input())
        l.append(x)
    a = l[:c]
    return (l[c::]+a[::])
    print(l)
n = int(input())
c = int(input())
splt(n,c)


```

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

Out[16]: `[4, 1, 2, 3]`

In [17]:

```
def largest(n):
    l = []
    for i in range(1,n):
        x = int(input())
        l.append(x)
    ii = max(l)
    return ii
n = int(input())
largest(n)
```

```
5
1
2
3
4
```

Out[17]:

```
4
```

In [20]:

```
def bub(n):
    l = []
    for i in range(n):
        x = int(input())
        l.append(x)
    for i in range(n-1):
        for j in range(0,n-i-1):
            if l[j] > l[j+1]:
                l[j],l[j+1]=l[j+1],l[j]
    return l
n = int(input())
bub(n)
```

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

Out[20]:

```
[1, 1, 2, 3, 4]
```

In [21]:

```python
def binary_search(arr, x):
    low = 0
    high = len(arr) - 1
    mid = 0
    while low <= high:
        mid = (high + low) // 2
        if arr[mid] < x:
            low = mid + 1
        elif arr[mid] > x:
            high = mid - 1
        else:
            return mid
    return -1

arr = []
for i in range(n):
    x = int(input())
    arr.append(x)
x = int(input())
result = binary_search(arr, x)

if result != -1:
    print("Element is present ", str(result))
else:
    print("Element is not present")
```

```
5
3
4
2
1
4
Element is present  2
```

In [22]:

```python
def mergeArrays(arr1, arr2, n1, n2):
    arr3 = [None]*(n1 + n2)
    i = 0
    j = 0
    k = 0
    while i < n1 and j < n2:
        if arr1[i] < arr2[j]:
            arr3[k] = arr1[i]
            k = k + 1
            i = i + 1
        else:
            arr3[k] = arr2[j]
            k = k + 1
            j = j + 1
    while i < n1:
        arr3[k] = arr1[i];
        k = k + 1
        i = i + 1
    while j < n2:
        arr3[k] = arr2[j];
        k = k + 1
        j = j + 1
    print("after")
    for i in range(n1 + n2):
        print(str(arr3[i]), end = " ")
arr1 = [1, 3, 5, 7]
n1 = len(arr1)

arr2 = [2, 4, 6, 8]
n2 = len(arr2)
mergeArrays(arr1, arr2, n1, n2);
```

```
after
1 2 3 4 5 6 7 8
```

In [40]:

```
import numpy as np
mat1 = []
mat2 = []
def creMatrix():
    r1 = int(input("ent r1: "))
    c1 = int(input("ent c1: "))
    r2 = int(input("ent r2: "))
    c2 = int(input("ent c2: "))

    for i in range(0,r1):
        l = []
        for j in range(0,c1):
            x = int(input())
            l.append(x)
        mat1.append(l)
    print(mat1)
    for i in range(0,r2):
        l = []
        for j in range(0,c2):
            x = int(input())
            l.append(x)
        mat2.append(l)
    print(mat2)
    le = len(mat2)
    #add
    res = np.zeros((le,le),dtype=int)
    res = res.tolist()
#     res = [[0,0],
#         [0,0]]
    for i in range(len(mat1)):
        for j in range(len(mat2[1])):
            res[i][j] = mat1[i][j] + mat2[i][j]
    print("add : ", res)

    #substract
    res = np.zeros((le,le),dtype=int)
    res = res.tolist()
#     res = [[0,0],
#         [0,0]]
    for i in range(len(mat1)):
        for j in range(len(mat2[1])):
            res[i][j] = mat1[i][j] - mat2[i][j]
    print("sub : ", res)

    #multiply
    res = np.zeros((le,le),dtype=int)
    res = res.tolist()
#     res = [[0,0],
#         [0,0]]
    for i in range(len(mat1)):
        for j in range(len(mat2[1])):
            res[i][j] = mat1[i][j] * mat2[i][j]
    print("mul : ", res)

creMatrix()

```

```
ent r1: 2
ent c1: 2
ent r2: 2
ent c2: 2
1
2
3
4
[[1, 2], [3, 4]]
1
2
3
4
[[1, 2], [3, 4]]
add :  [[2, 4], [6, 8]]
sub :  [[0, 0], [0, 0]]
mul :  [[1, 4], [9, 16]]
```

In [8]:

```
def Func():

    list = [["ABC", "EFG"], ["HIJ", "KLM"]]
    res = []
    n = 0
    while n != len(list):
        temp = ''
        for i in list:
            try: temp = temp + i[n]
            except IndexError: pass
        res.append(temp)
        n = n + 1
    res = [ele for ele in res if ele]
    print("Column Concat : " + str(res))
Func()
```

```
Column Concat : ['ABCHIJ', 'EFGKLM']
```

In [ ]:

```

```