In the name of God

استاد : دکتر تیموری

دانشجو : توحید حقیقی سیس

شماره دانشجویی : 830598021

موضوع : تمرین اول

تمرین اول :

۱. ماتریس C با ابعاد M×n و بردار سطری c با طول n را در نظر بگیرید. فرض کنید عناصر هر دو تنها دارای یکی از مقادیر صفر و یا یک هستند. تابعی بنویسید که با گرفتن ماتریس C و بردار c، اندیس نزدیکترین سطر ماتریس C به بردار c را (با معیار فاصله همینگ) در خروجی برگرداند.

در ابتدا یک تابع برای گرفتن ماتریس ها از ورودی را مینویسیم .

ما برای این سوال یک ماتریس 2 بعدی و یک وکتور نیاز داریم برای این کار به صورت زیر عمل میکنیم .

```
#-------------------- تابع وارد کردن ماتریس --------------------
def input_matrix(m, n,msg="Please enter value for M_{},{} : "):
    l_mat=[]
    for i in range(m):
        r = []
        for j in range(n):
            e = int(input(msg.format(i, j)))
            r.append(e)
        l_mat.append(r)
    return np.array(l_mat)
```

در داخل 2 فور تو در تو محتوای ماتریس 2 بعدی را میگیرد و اگر بخواهیم وکتور بگیریم یکی از ارگومان های ان را 1 میگذاریم .

```
import numpy as np

#-------------------- تابع پیدا کردن تابع همیک --------------------
def calc_hamming_distancce(a, b):
    a.reshape(b.shape)
    return np.count_nonzero(a != b)
```

تابع همینگ یکی از تابع های مشهور است و تعداد ان هایی که محتوای ان ها برابر نیست را محاسبه میکند .

```
    return np.array(i_mat)

#------------------- تمرین اول ----------------------------------------
m = int(input("Please input m :"))
n = int(input("Please input n :"))
C = input_matrix(m, n, msg="Please enter value for C_{},{} : ")
print("*"*10)
c = input_matrix(1, n, msg="Please enter value for c_{},{} : ")
print("*"*10)
nni = np.argmin(np.apply_along_axis(lambda x:calc_hamming_distancce(x, c), axis
print("Nearest Neighbour Index : {}".format(nni))


#-----------------------------------------------------------------------
```

و از توابع numpy برای محاسبه مقدار فاصله هینگ بین سطر های ماتریس و وکتور را مییابد و در اخر کمترین ان را بر میگرداند .

تمرین 2 :

۲. تابعی در MATLAB بنویسید که با گرفتن عدد صحیح N تمام شمارنده‌های آن را به ترتیب صعودی برگرداند. به عنوان مثال، این تابع باید با ازای عدد 6 بردار [1,2,3,6] را در خروجی خود بدهد.

در ابتدا مقدار را از ورودی میگیریم و بعد بر اساس تابع زیر تمام زیر مجموعه های ان را محاسبه میکنیم .

```python
#-------------- تمرین دوم ------------------------------
def find_factors(N):
    factors = {1,N}
    for i in range(2,int(sqrt(N))+1):
        if N%i == 0:
            factors.update((i,N//i))
    return factors

N = int(input("Please enter value for N : "))
print("Factors of {} = {}".format(N, find_factors(N)))
```

و خروجی ان نیز به صورت زیر است :

```
Please enter value for N : 12
Factors of 12 = {1, 2, 3, 4, 6, 12}
PS E:\hw1> & C:/Users/tohid-pc/AppData/Local/Programs/Pytho
Please enter value for N : 50
Factors of 50 = {1, 2, 5, 10, 50, 25}
```

تمرین 3 :

۳. تابعی بنویسید که کوچکترین N عدد اول ممکن را تولید نماید.

توجه: برای محاسبه از تابع آماده primes استفاده نکنید.

برای حل این مسئله ابتدا باید n را از ورودی گرفته و بعد از عدد 1 شروع و به ترتیب بالا رویم تا تمام اعداد اول را به دست اوریم .

برای محاسبه تابع اول بودن ان را به صورت زیر پیاده کرده ام .

```python
def is_prime(a):
    if((a<=1) or (a%1>0)):
        return False
    for i in range(2, a//2):
        if(a%i==0):
            return False
    return True
```

تابع محاسبه اول بودن به صورت بالا است .

```python
    return True

N = int(input("Please enter value for N : "))
prime_set = set()
i = 0
while(N>0):
    if(is_prime(i)):
        prime_set.add(i)
        N -= 1
    i += 1
print("{} first prime numbers = {}".format(len(prime_set), prime_s
```

و خروجی ان نیز به صورت زیر میشود .

```
Please enter value for N : 12
12 first prime numbers = {2, 3, 4, 5, 7, 11, 13, 17, 19, 23, 29, 31}
PS E:\hw1>
```

تمرین 4 :

۴. تابعی بنویسید که تمامی جواب‌های صحیح معادله $x_1+x_2+...+x_n=k$ را با فرض $x_i \geq 0$ در ماتریسی با n ستون بدهد. ورودی‌های این تابع k و n هستند.

در این سوال باید k , n رو به صورت ورودی گرفته و ان را به تابع calc_x پاس میدهیم . این تابع تمام حالت های مختلف n و k را به دست اورده و در معادله قرار میدهد تا جواب ها را پیدا کند .

کد این قسمت به صورت زیر عمل میکند :

```
from itertools import chain, combinations, permutations

def get_subsets(l):
    return chain(*[combinations(l,i + 1) for i,a in enumerate(l)])

def get_k_subsets(l, k):
    s_l = sorted(l)
    return set([tuple(i) for i in combinations(get_subsets(l),k)
                if sorted(chain(*i)) == s_l])

# x_1 + x_2 + ... + x_n = k
def calc_x(k, n):
    l_k = [1] * k
    k_subsets = get_k_subsets(l_k, n)
    answers = []
    for el in k_subsets:
        for per in set(permutations(el)):
            ans = list(map(sum, per))
            answers.append(ans)
    return np.array(answers)

k = int(input("Enter value for k :"))
n = int(input("Enter value for n :"))
print("answers = {}".format(calc_x(k, n)))
```

و خروجی ان نیز به صورت زیر خواهد بود :

```
Enter value for k :3
Enter value for n :3
answers = [[1 1 1]]
PS E:\hw1> & C:/Users/tohid-pc/AppData/Local/Programs/Python/Python38-32/python.exe e:/hw1/Homework.py
Enter value for k :12
Enter value for n :5
```