

تمرین شماره 1 شبیه سازی و مدلسازی | دکتر تیموری

تمرین های اولیه کدنویسی

صادق رحمتی | 830598026

کد تمرین شماره ۱ :

```
from math import sqrt
import numpy as np
def calc_hamming_distance(a, b):
    a.reshape(b.shape)
    return np.count_nonzero(a != b)
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)
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_distance(x, c), axis=1,
arr=C))
print("Nearest Neighbour Index : {}".format(nni))
```

خروجی تمرین شماره ۱ :

```
sadiq@sadiq-laptop ~/w/w sm/hw1 python3 q1.py
Please input m :2
Please input n :3
Please enter value for C_0,0 : 1
Please enter value for C_0,1 : 2
Please enter value for C_0,2 : 3
Please enter value for C_1,0 : 4
Please enter value for C_1,1 : 5
Please enter value for C_1,2 : 6
*****
Please enter value for c_0,0 : 1
Please enter value for c_0,1 : 5
Please enter value for c_0,2 : 6
*****
Nearest Neighbour Index : 1
```

کد تمرین شماره ۲ :

```
from math import sqrt
import numpy as np
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)))
```

خروجی تمرین شماره ۲ :

```
sadiq@sadiq-laptop ~/w/w sm/hw1 python3 q2.py
Please enter value for N : 24
Factors of 24 = {1, 2, 3, 4, 6, 8, 12, 24}
```

کد تمرین شماره ۳ :

```
from math import sqrt
import numpy as np
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
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_set))
```

خروجی تمرین شماره ۳ :

```
sadiq@sadiq-laptop ~/w/w sm/hw1 python3 q3.py
Please enter value for N : 12
12 first prime numbers = {2, 3, 4, 5, 7, 11, 13, 17, 19, 23, 29, 31}
```

کد تمرین شماره ۴ :

```
from math import sqrt
import numpy as np
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)))
```

خروجی تمرین شماره ۴ :

```
sadiq@sadiq-laptop ~/w/w sm/hw1 python3 q4.py
Enter value for k :5
Enter value for n :3
answers = [[1 1 3]
 [1 3 1]
 [3 1 1]
 [2 2 1]
 [2 1 2]
 [1 2 2]]
```