문제 1. 요리사의 고민 ( # 2961 )

N개의 재료가 주어집니다.

각각의 재료에는 단맛과 짠맛이 있습니다.

N개의 줄에 재료의 단맛 A와 짠맛 B가 주어집니다.

요리에 재료를 추가할 때마다, 재료의 단맛은 기존의 단맛에 더해지고, 짠맛은 기존의 짠맛에 곱해집니다.

단맛과 짠맛의 차이가 가장 적은 요리를 만들었을 때,

단맛과 짠맛의 차이를 출력하는 프로그램을 작성하세요.

4

1 7

2 6

3 8

4 9

1

문제 2. 요리사의 복잡한 고민 ( #19942 )

N개의 재료가 주어집니다.

그 다음 채워야 하는 영양소 a,b,c,d가 주어집니다.

각각의 재료에는 영양소가 있습니다.

N개의 줄에 재료의 단백질 A와 지방 B 탄수화물 C 비타민 D 가격 E 가 주어집니다.

요리에 재료를 추가할 때마다, 영양소는 기존의 영양소에 더해집니다.

가장 싼 가격으로 원하는 영양소를 모두 채웠다고 했을 때, 그 가격을 계산하는 프로그램을 작성하시오.

6

100 70 90 10

30 55 10 8 100

60 10 10 2 70

10 80 50 0 50

40 30 30 8 60

60 10 70 2 120

20 70 50 4 4

134

문제 3. 상담 ( #14501 )

상담원 정우는 곧 퇴사를 하려고 합니다.

퇴사 날은 N+1일 이라서, 남은 N일 동안 상담을 해서 돈을 많이 벌어서 나가고 싶습니다.

가장 돈을 많이 받도록 상담을 골라서 신청했을 때, 그 금액을 출력하는 프로그램을 작성하세요.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 1일 | 2일 | 3일 | 4일 | 5일 | 6일 | 7일 |
| T | 4 | 1 | 3 | 1 | 2 | 3 | 2 |
| P | 10 | 20 | 30 | 40 | 35 | 150 | 10 |

7

4 10

1 20

3 30

1 40

2 35

3 150

2 10

95

문제 4. 냅색 ( #12865 )

물건의 수 A와 배낭의 무게 B가 주어집니다.

순서대로 물건의 무게 W와 가치 V가 주어집니다.

배낭에 가장 가치있게 담았을 경우의 가치를 출력하는 프로그램을 작성하세요.

4 7

6 13

4 8

3 6

5 12

14

정답코드

(1)

def recur(idx, sin, sun, use):

global answer

if idx == n:

if use > 0:

answer = min(answer, abs(sin - sun))

return

recur(idx + 1, sin \* foods[idx][0], sun + foods[idx][1], use+1)

recur(idx + 1, sin, sun, use)

n = int(input())

foods = [() for \_ in range(n)]

for i in range(n):

a, b = map(int, input().split())

foods[i] = (a, b)

answer = 1e9

recur(0, 1, 0, 0)

print(answer)

(2)

def recur(idx,p,f,c,v,price):

global answer

global used

global answer\_used

if p >= protein and f >= fat and c >= carbo and v >= vitamin:

if answer > price:

answer = min(answer, price)

answer\_used = []

for i in used:

answer\_used.append(i)

if idx == n:

return

used.append(idx+1)

recur(idx+1, p+ing[idx][0], f +ing[idx][1], c + ing[idx][2], v + ing[idx][3], price+ing[idx][4])

used.pop()

recur(idx+1,p,f,c,v,price)

n = int(input())

protein, fat, carbo, vitamin = map(int,input().split())

ing = [[] for \_ in range(n)]

for i in range(n):

a,b,c,d,e = map(int,input().split())

ing[i] = [a,b,c,d,e]

answer = 1e9

used = []

answer\_used = []

recur(0,0,0,0,0,0)

answer\_used.sort()

if answer\_used:

print(answer)

print(\*answer\_used)

else:

print(-1)

(3)  
def recur(idx, result):

global answer

if idx > n:

if idx > n+1: return

answer = max(answer, result)

return

recur(idx + table[idx][0], result + table[idx][1])

recur(idx + 1, result)

n = int(input())

table = [[] for \_ in range(n+1)]

for i in range(n):

a, b = map(int, input().split())

table[i+1] = [a, b]

# print(table)

answer = 0

recur(1, 0)

print(answer)

(4)

다음 강의로!