

Олимпиада «Ломоносов» по информатике
2024-2025 учебный год. Заключительный этап
Работа участника с id заявки 1430676, логином inf25f_367

Сводный итог по всем задачам в проверяющей системе

RunID	Time	Username	Prob	Lang	Result	Tests	Score
364	3:58:25	inf25f_367	3	python3	Partial solution	3	5
351	3:56:15	inf25f_367	5	python3	Partial solution	0	0
120	1:51:11	inf25f_367	2	python3	OK	28	100
N/A	N/A	inf25f_367	5	N/A	N/A	0	0
N/A	N/A	inf25f_367	1	N/A	N/A	0	0

105 технических баллов

30 итоговых баллов

Посылка по задаче 1

Посылка по задаче 1 не было отправлено.

Посылка по задаче 2

```
[1]
[2]
[3]     b = ["A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P",
"Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z"]
[4]     l_offset = 26
[5]     d = {
[6]         "^": 1,
[7]         "~": 2,
[8]         "_": 3
[9]     }
[10]    def translate(str_):
[11]        out = 0
[12]        r = 0
[13]        pr = 0
[14]        n_amplifier = 0
[15]        offset = False
[16]        for s_i in range(len(str_)-1, -1, -1):
[17]            if pr == 1:
[18]                pr = 0
[19]                n_amplifier = 0
[20]                offset = False
[21]                continue
[22]            if str_[s_i] in d.keys():
[23]                pr += 1
[24]                offset = True
[25]                n_amplifier = d[str_[s_i]]
[26]                n = b.index(str_[s_i - int(offset)].upper()) + l_offset if str_[s_i -
int(offset)].lower() == str_[s_i - int(offset)] else b.index(str_[s_i -
int(offset)].upper())
[27]                out += (52 ** r) * (n + 52 * n_amplifier)
[28]                r += 1
[29]
[30]        return out
[31]
[32]    N = int(input())
[33]
[34]    l = [translate(input()) for i in range(N)]
[35]    s_l = sorted(l, reverse=True)
[36]    ind = []
[37]    for i in range(len(l)):
[38]        if s_l[i] != l[i]:
[39]            ind.append(str(i+1))
[40]    print(" ".join(ind))
[41]
[42]
```

Посылка по задаче 3

```
[1] N, T, M = list(map(int, input().split()))
[2] points = list(map(int, input().split()))
[3] places = {}
[4] ways = []
[5] ways_time = []
[6] for i in range(N):
[7]     row = list(map(int, input().split()))
[8]     for j in range(N):
[9]         if i == j:
[10]             continue
[11]         if i in places.keys():
[12]             places[i].append(j)
[13]         else:
[14]             places[i] = [j]
[15]         if not [j, i] in ways and not [i, j] in ways:
[16]             ways.append([i, j])
[17]             ways_time.append(row[j])
[18] for road_i in range(M):
[19]     i, j, row_j = list(map(int, input().split()))
[20]     if [i-1, j-1] in ways:
[21]         ways_time[ways.index([i-1, j-1])] = min(ways_time[ways.index([i-1, j-1])], row_j)
[22]     elif [j-1, i-1] in ways:
[23]         ways_time[ways.index([j-1, i-1])] = min(ways_time[ways.index([j-1, i-1])], row_j)
[24] outt = []
[25] def f(s=0, e=0, p=0, t=0, path=None):
[26]     global outt
[27]     if path is None:
[28]         path = []
[29]     if s == e and t != 0 and t <= T:
[30]         return [p, path]
[31]     elif t >= T:
[32]         return [-1, None]
[33]     else:
[34]         for i in places[s]:
[35]             if not(i in path) or i == 0:
[36]                 outt.append(f(s=i, p=p+points[s], t=(t+ways_time[ways.index([i, s] if [i, s] in ways else [s, i]))), p
[37]             if t == 0 and s == 0:
[38]                 return outt
[39]             else:
[40]                 return [-1, None]
[41] answ = f()
[42] while [-1, None] in answ:
[43]     answ.remove([-1, None])
[44] if not answ:
[45]     print(1)
[46]     print(1)
[47] else:
[48]     a = sorted(answ, key=lambda x: x[0], reverse=True)[0][1]
[49]     print(len(a))
[50]     print(" ".join(list(map(str, a))))
```

Посылка по задаче 4

Посылка по задаче 1 не было отправлено.

Посылка по задаче 5

```
[1] M, N, K = list(map(int, input().split()))
[2] d = {0:set()}
[3] for contact_i in range(N-1):
[4]     row = list(map(int, input().split()))
[5]     d[row[1]-1] = d[row[0]-1] | {row[2]}
[6] first, second = list(map(int, input().split()))
[7]
[8]
[9] mmax = 0
[10] r = sorted(list(d[first-1] | d[second-1]))
[11] x_0 = 1
[12] i = 0
[13] for otr in range(len(r)+1):
[14]
[15]     if x_0 == 1:
[16]         mmax = max(mmax, r[i]-x_0)
[17]         x_0 = r[i]
[18]         i += 1
[19]     elif i == len(r):
[20]         mmax = max(mmax, M-r[i-1])
[21]     else:
[22]         mmax = max(mmax, r[i]-x_0-1)
[23]         x_0 = r[i]
[24]         i += 1
[25] print(mmax)
```