Олимпиада «Ломоносов» по информатике
2024-2025 учебный год. Заключительный этап
Работа участника с id заявки 1383974, логином inf25f_342

Сводный итог по всем задачам в проверяющей системе

RunID	Time	Username	Prob	Lang	Result	Tests	Score
361	3:58:02	inf25f_342	4	python3	Partial solution	0	0
237	3:12:26	inf25f_342	3	python3	Partial solution	19	80
169	2:35:05	inf25f_342	5	python3	Partial solution	1	0
96	1:26:50	inf25f_342	2	python3	OK	28	100
18	0:36:24	inf25f_342	1	python3	Partial solution	12	32

212 технических баллов
61 итоговый балл

Посылка по задаче 1

```
[1] nach = int(input())
[2] kon = int(input())
[3] predpoc = [[nach]]
[4]
[5] a = 0
[6] if nach == kon:
[7]     a = 1
[8]     itog = []
[9] j = 0
[10] while a != 1:
[11]     curpoc = []
[12]     for n in predpoc:
[13]         if n[-1] % 2 == 0:
[14]             n2 = n[:]
[15]             if n[-1] // 2 == kon:
[16]                 a = 1
[17]                 itog = n[:]
[18]                 break
[19]             n2.append(n[-1] // 2)
[20]             curpoc.append(n2)
[21]         else:
[22]             n2 = n[:]
[23]             if n[-1] * 3 + 1 == kon:
[24]                 a = 1
[25]                 itog = n[:]
[26]                 break
[27]             n2.append(n[-1] * 3 + 1)
[28]             curpoc.append(n2)
[29]         if (n[-1] - 1) % 3 == 0:
[30]             n2 = n[:]
[31]             if (n[-1] - 1) // 3 == kon:
[32]                 a = 1
[33]                 itog = n[:]
[34]                 break
[35]             n2.append((n[-1] - 1) // 3)
[36]             curpoc.append(n2)
[37]         n2 = n[:]
[38]         if n[-1] * 2 == kon:
[39]             a = 1
[40]             itog = n[:]
[41]             break
[42]         n2.append(n[-1] * 2)
[43]         curpoc.append(n2)
[44]
[45]     predpoc = curpoc[:]
[46] print(len(itog))
[47] if len(itog) >= 2:
[48]     print(*itog[1:])
```

Посылка по задаче 2

```
[1]
[2] def per(n):
[3]
[4]     i = 0
[5]     itog = []
[6]     while i in range(len(n)-1):
[7]         osn = 0
[8]         j = 0
[9]         if n[i+1] in a:
[10]            osn = a.index(n[i+1])
[11]            j += 1
[12]
[13]        itog.append(52*osn +alph.index(n[i]))
[14]        i += 1 + j
[15]    if n[-1] not in a:
[16]        itog.append(alph.index(n[-1]))
[17]    it = 0
[18]    itog = itog[::-1]
[19]    for j in range(len(itog)):
[20]        it += (52**j)*itog[j]
[21]    return it
[22]
[23]
[24]
[25] a = ["", "^", "~", "_"]
[26] alph = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
[27] c = int(input())
[28] i1 = -1
[29] i2 = -1
[30] pr = float("inf")
[31] k = []
[32] for i in range(c):
[33]     cur = per(input())
[34]     k.append(cur)
[35] k1 = k[:]
[36] k.sort(reverse=True)
[37] for i in range(len(k)):
[38]     if k[i] != k1[i]:
[39]         print(i+1, end=" ")
[40]
```

Посылка по задаче 3

```
[1] def sumway(way):
[2]     itog = 0
[3]     for i in set(way):
[4]         itog += gold[i]
[5]     return itog
[6]
[7] def findway(n, ost, way):
[8]     if ost < 0:
[9]         return [0, []]
[10]    if way[-1] == 0 and len(way) != 1:
[11]        return [-sumway(way), way]
[12]    sp = []
[13]    for u in range(allp):
[14]        if u != n:
[15]            sp.append(findway(u, ost - time[n][u], way + [u]))
[16]    return min(sp)
[17]
[18] allp, maxtime, countroad = list(map(int, input().split()))
[19] time = []
[20] gold = list(map(int, input().split()))
[21] for i in range(allp):
[22]    time.append(list(map(int, input().split())))
[23] for i in range(countroad):
[24]    p1, p2, timec = list(map(int, input().split()))
[25]    time[p1-1][p2-1] = min(timec, time[p1-1][p2-1])
[26]    time[p2-1][p1-1] = min(timec, time[p2-1][p1-1])
[27] itog = list(map(lambda s: s+1,list(set(findway(0, maxtime, [0])[1]))))
[28] itog.sort()
[29]
[30] if itog:
[31]    print(len(itog))
[32]    print(*itog)
[33] else:
[34]    print(1)
[35]    print(gold[0])
```

Посылка по задаче 4

```
[1] for i in stdin:
[2]     i = i.replace("[", "(")
[3]     i = i.replace("]", ")")
[4]     before, after = i.split("=")
[5]     itog = ""
[6]     itog2 = ""
[7]     for i in set(before):
[8]         itog = itog + i + "+"
[9]         itog2 = itog2 + i
[10]    itog = itog[:-1]
[11]    print(itog + "=" + itog2)
[12]
[13]
[14]
[15]
```

Посылка по задаче 5

```
[1] lenrnk, pac, zap = list(map(int,input().split()))
[2] it = [[]]*pac
[3] for i in range(pac-1):
[4]     p0, p1, num =  list(map(int,input().split()))
[5]     p = it[p0-1][:]
[6]     p.append(num)
[7]     it[p1-1] = p
[8]
[9] for l in range(zap):
[10]    m = 0
[11]    cur = [0, pac-1]
[12]    z1, z2 = list(map(int,input().split()))
[13]    for k in it[z1-1]:
[14]        cur.append(k)
[15]    for k in it[z2-1]:
[16]        cur.append(k)
[17]    cur.sort()
[18]    for j in range(len(cur)-1):
[19]        m = max(m, cur[j+1]-cur[j])
[20] print(m)
```