

Олимпиада «Ломоносов» по информатике
2023-2024 учебный год. Заключительный тур
Работа участника с id заявки 1198225, логином inf24f_234

Сводный итог по всем задачам в проверяющей системе

Run ID	Time	User name	Problem	Language	Result	Tests	Score
14	0:11:07	inf24f_234	1	python3	OK	28	100
152	1:02:21	inf24f_234	2	python3	OK	28	100
293	1:57:00	inf24f_234	3	python3	Partial solution	24	84
389	2:35:15	inf24f_234	5	python3	OK	22	100
508	3:20:35	inf24f_234	6	python3	Partial solution	7	60
444	технических балла						
74	итоговых балла						

Посылка по задаче 1

```
[1] t0, t1, t2, t3 = 0, 0, 0, 1
[2] tetra_nums = []
[3] for i in range(34):
[4]     t0, t1, t2, t3 = t1, t2, t3, (t0 + t1 + t2 + t3)
[5]     tetra_nums.append(t3)
[6] tetra_nums = tetra_nums[::-1]
[7] n = int(input())
[8] col_ob = 0
[9] for i in range(n):
[10]     a = int(input())
[11]     col = 0
[12]     j = 0
[13]     while a > 0:
[14]         if a >= tetra_nums[j]:
[15]             col += 1
[16]             a -= tetra_nums[j]
[17]             j += 1
[18]     if col % 2 == 0:
[19]         col_ob += 1
[20] print(col_ob)
```

Посылка по задаче 2

```
[1] line = input()
[2] queue = [2]
[3] doli = {'B': '0', 'C': '0', 'D': '0', 'G': '0', 'O': '0', 'R': '0', 'V': '0', 'W': '0', 'Y': '0'}
[4] tek = 0
[5] if len(line) == 1:
[6]     print(line)
[7]     print(1.0)
[8]     quit()
[9] for i in range(len(line)):
[10]    if line[i] == 'Q':
[11]        queue[-1] -= 1
[12]        queue.append(8)
[13]        tek += 3
[14]    else:
[15]        if len(doli[line[i]]) < (tek + 1):
[16]            doli[line[i]] = doli[line[i]] + '0' * ((tek + 1) - len(doli[line[i]]))
[17]        doli[line[i]] = doli[line[i]][:tek] + str(int(doli[line[i]][tek]) + 1) + doli[line[i]][tek + 1:]
[18]        if doli[line[i]][tek] == '2':
[19]            tek0 = tek
[20]            while doli[line[i]][tek0] == '2':
[21]                doli[line[i]] = doli[line[i]][:tek0] + '0' + doli[line[i]][tek0 + 1:]
[22]                tek0 -= 1
[23]            doli[line[i]] = doli[line[i]][:tek0] + str(int(doli[line[i]][tek0]) + 1) + doli[line[i]][tek0 + 1:]
[24]        queue[-1] -= 1
[25]        while queue[-1] == 0:
[26]            queue.pop(-1)
[27]            tek -= 3
[28]    mx_b = 'B'
[29]    for key in doli.keys():
[30]        if doli[key] > doli[mx_b]:
[31]            mx_b = key
[32]    num = doli[mx_b]
[33]    num = num[0] + '.' + num[1:]
[34]    while num[-1] == '0':
[35]        num = num[:-1]
[36]    print(mx_b)
[37]    print(num)
```

Посылка по задаче 3

```
[1] def perevod(line):
[2]     if line == '[]':
[3]         return '0' * 100
[4]     else:
[5]         line2 = ''
[6]         i = 0
[7]         while i < len(line):
[8]             if line[i] == 'i':
[9]                 if (i + 1) < len(line) and line[i + 1] == '(':
[10]                    if (i + 2) < len(line) and line[i + 2] == '(':
[11]                        line2 += '3'
[12]                        i += 3
[13]                    else:
[14]                        line2 += '2'
[15]                        i += 2
[16]                else:
[17]                    line2 += '1'
[18]                    i += 1
[19]            elif line[i] == 'I':
[20]                if (i + 1) < len(line) and line[i + 1] == '(':
[21]                    if (i + 2) < len(line) and line[i + 2] == '(':
[22]                        line2 += '6'
[23]                        i += 3
[24]                    else:
[25]                        line2 += '5'
[26]                        i += 2
[27]                else:
[28]                    line2 += '4'
[29]                    i += 1
[30]            elif line[i] == 'J':
[31]                if (i + 1) < len(line) and line[i + 1] == ')':
[32]                    if (i + 2) < len(line) and line[i + 2] == ')':
[33]                        line2 += '9'
[34]                        i += 3
[35]                    else:
[36]                        line2 += '8'
[37]                        i += 2
[38]                else:
[39]                    line2 += '7'
[40]                    i += 1
[41]            elif line[i] == 'j':
[42]                if (i + 1) < len(line) and line[i + 1] == ')':
[43]                    if (i + 2) < len(line) and line[i + 2] == ')':
[44]                        line2 += 'C'
[45]                        i += 3
[46]                    else:
[47]                        line2 += 'B'
[48]                        i += 2
[49]                else:
[50]                    line2 += 'A'
[51]                    i += 1
[52]        line2 = line2[::-1]
[53]        line2 = '0' * (100 - len(line2)) + line2
[54]        return line2
```

```

[55]
[56]
[57] def perevod_obratno(index):
[58]     per = {'12': 'j)', '11': 'j)', '10': 'j', '9': 'j))', '8': 'j)', '7': 'j', '6': 'I(', '5': 'I(', '4': 'I', '3': 'i(', '2': 'i(', '1': 'i'}
[59]     perv = index // 144
[60]     vtor = (index % 144) // 12
[61]     tret = index % 12
[62]     if tret == 0:
[63]         vtor -= 1
[64]         tret = 12
[65]     if vtor == 0:
[66]         if perv == 0:
[67]             return per[str(tret)]
[68]         else:
[69]             perv -= 1
[70]             vtor = 12
[71]             if perv == 0:
[72]                 return per[str(tret)] + per[str(vtor)]
[73]     return per[str(tret)] + per[str(vtor)] + per[str(perv)]
[74]
[75]
[76] n = int(input())
[77] mx = '0' * 100
[78] mx_index = 0
[79] mn = 'C' * 100
[80] mn_index = 0
[81] for i in range(n):
[82]     line = input()
[83]     line2 = perevod_obratno(line)
[84]     if line2 >= mx:
[85]         mx = line2
[86]         mx_index = i + 1
[87]     if line2 <= mn:
[88]         mn = line2
[89]         mn_index = i + 1
[90] if mn_index == mx_index:
[91]     print(perevod_obratno(n - 1))
[92]     print(perevod_obratno(n))
[93] elif mn_index < mx_index:
[94]     print(perevod_obratno(mn_index))
[95]     print(perevod_obratno(mx_index))
[96] else:
[97]     print(perevod_obratno(mx_index))
[98]     print(perevod_obratno(mn_index))

```

Посылок по задаче 4 не было

Посылка по задаче 5

```
[1] def main():
[2]     transparant = input()
[3]     needed_transparant = input()
[4]     cur = {}
[5]     needed = {}
[6]     for i in range(33, 127):
[7]         cur.update([(chr(i), 0)])
[8]         needed.update([(chr(i), 0)])
[9]     for i in range(len(needed_transparant)):
[10]        needed[needed_transparant[i]] += 1
[11]    mn_peace = ''
[12]    cur_peace = ''
[13]    perv = False
[14]    need_letter = ''
[15]    for i in range(0, len(transparant)):
[16]        if perv == False:
[17]            cur[transparant[i]] += 1
[18]            cur_peace += transparant[i]
[19]            if i >= (len(needed_transparant) - 1):
[20]                perv = True
[21]                for key in needed.keys():
[22]                    if cur[key] < needed[key]:
[23]                        perv = False
[24]                        break
[25]                if perv:
[26]                    while cur[cur_peace[0]] > needed[cur_peace[0]]:
[27]                        cur[cur_peace[0]] -= 1
[28]                        cur_peace = cur_peace[1:]
[29]                    mn_peace = cur_peace
[30]                    need_letter = cur_peace[0]
[31]                    cur[cur_peace[0]] -= 1
[32]                    cur_peace = cur_peace[1:]
[33]            else:
[34]                cur[transparant[i]] += 1
[35]                cur_peace += transparant[i]
[36]                if transparant[i] == need_letter:
[37]                    while cur[cur_peace[0]] > needed[cur_peace[0]]:
[38]                        cur[cur_peace[0]] -= 1
[39]                        cur_peace = cur_peace[1:]
[40]                    if len(cur_peace) < len(mn_peace):
[41]                        mn_peace = cur_peace
[42]                        need_letter = cur_peace[0]
[43]                        cur[cur_peace[0]] -= 1
[44]                        cur_peace = cur_peace[1:]
[45]    print(mn_peace)
[46]
[47]
[48] if __name__ == '__main__':
[49]     main()
```

Посылка по задаче 6

```
[1] import sys
[2] csv = sys.stdin.read().split('\n')[:-1]
[3] mdict = {}
[4] for i in range(len(csv)):
[5]     line = csv[i].split(';')
[6]     if line[0] not in mdict.keys():
[7]         if all([elem == 'OK' for elem in line[1:]] or not any([elem == 'OK' for elem in line[1:]]):
[8]             mdict.update([(line[0], [len(line) - 1, 1, []])])
[9]         else:
[10]            mdict.update([(line[0], [len(line) - 1, 2, []])])
[11]            for j in range(1, len(line) - 1):
[12]                if line[j] == 'OK':
[13]                    if line[j + 1] != 'OK':
[14]                        mdict[line[0]][2].append(j)
[15]                    else:
[16]                        if line[j + 1] == 'OK':
[17]                            mdict[line[0]][2].append(j)
[18]            else:
[19]                mdict[line[0]] = [max(mdict[line[0]][0], len(line) - 1), mdict[line[0]][1], mdict[line[0]][2]]
[20]                for j in range(1, len(line) - 1):
[21]                    if line[j] == 'OK':
[22]                        if line[j + 1] != 'OK':
[23]                            if j not in mdict[line[0]][2]:
[24]                                mdict[line[0]][1] += 1
[25]                                mdict[line[0]][2].append(j)
[26]                    else:
[27]                        if line[j + 1] == 'OK':
[28]                            if j not in mdict[line[0]][2]:
[29]                                mdict[line[0]][1] += 1
[30]                                mdict[line[0]][2].append(j)
[31] for key in sorted(mdict.keys()):
[32]     print(mdict[key][0], mdict[key][1])
[33]
```