МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. ЛОМОНОСОВА

ОЛИМПИАДНАЯ РАБОТА

Наименование олимпиады школьников: **«Ломоносов»**

Профиль олимпиады: **Информатика**

ФИО участника олимпиады: **Мишин Сергей Максимович**

Класс: **9 класс**

Технический балл: **88**

Дата проведения: **17 марта 2022 г.**

Результаты проверки:

Оценка участника строится в 2 этапа:

1. оценка за задание - рассчитывается путем запуска тестов и определения правильности работы программы на тестах, до 100 баллов по каждой задаче;
2. нормализация оценки - полученная сумма делится на 2.9.

Оценки за задания:

№	1	2	3	4	5
Оценка	94	90	0	72	0

Задание 1. Попытка 1.

```
def litval(c: str) -> int:
    return int(c) if '0' <= c <= '9' else 10 + ord(c) - ord('A')



def fact(n: int) -> int:
    ret = 1
    for i in range(1, n + 1):
        ret *= i
    return ret



def convert(s: str) -> int:
    ret = 0
    for mult, num in enumerate(reversed(s)):
        ret += litval(num) * fact(mult + 1)
    return ret



def main():
    k = int(input())
    n = int(input())
    seq = [input() for _ in range(n)]
    kfact = fact(k)
```

```
    for idx, num in enumerate(reversed(seq)):
        mean_num = num.lstrip('0')
        mean_num = mean_num if mean_num else '0'
        to_test = mean_num[len(mean_num) - k + 1:]
        if not convert(to_test) % kfact:
            print(mean_num)
            print(n - idx)
            return

    print('-1')



main()
```

Задание 1. Попытка 2.

```
def litval(c: str) -> int:
    return int(c) if '0' <= c <= '9' else 10 + ord(c) - ord('A')



def fact(n: int) -> int:
    ret = 1
    for i in range(1, n + 1):
        ret *= i
    return ret



def convert(s: str) -> int:
    ret = 0
    for mult, num in enumerate(reversed(s)):
        ret += litval(num) * fact(mult + 1)
    return ret



def main():
    k = int(input())
    n = int(input())
    seq = [input() for _ in range(n)]
    kfact = fact(k)
```

```
    # PћC,PIPµC, PsPrPëPS Pë PjP°PєCЃPëPjP°P»CЊPSPs P±P»PëP·PsPє Pє
PєPsPSC†Cґ -> PëPrC‘Pj CЃ PєPsPSC†P°

    for idx, num in enumerate(reversed(seq)):

        mean_num = num.lstrip('0')

        mean_num = mean_num if mean_num else '0'

        # P’CЃPµ C†PëC„CЂC‹ PїPsCЃP»Pµ k-PsP№ CЃ PєPsPSC†P° PєCЂP°C,PSC‹ k!

        to_test = mean_num[len(mean_num) - k + 1:]

        if not convert(to_test) % kfact:

            print(mean_num)

            print(n - idx)

            return


    print('-1')



main()
```

Задание 2. Попытка 1.

```
def litval(c: str) -> int:
    return int(c) if '0' <= c <= '9' else 10 + ord(c) - ord('A')



def use(rem: dict, num: str):
    rem[num] -= 1
    if not rem[num]:
        rem.pop(num)



def gen_usable(usable: int, rem: dict, func: callable) -> str:
    pot_usable = [key for key in rem if litval(key) <= usable]
    return func(pot_usable, key=litval) if pot_usable else None



def main():
    input()
    seq = input()

    avail_chars = {}
    for char in seq:
        if char.isdigit() or (char.isalpha() and char.isupper()):
            avail_chars[char] = avail_chars.get(char, 0) + 1
```

```
    # РџPsCЃC,CЂPsPëPj C‡PëCЃP»Ps PSP°PëP±PsP»CЊC€РµР№ РґР»РëPSC‹
    num: list[str] = []
    while (curr := gen_usable(len(num) + 1, avail_chars, min)) \
            is not None and len(num) < 35:
        num.append(curr)
        use(avail_chars, curr)
    num = num[::-1]


    # PJPIPµP»PëC‡PëPj P·PSP°C‡PëPjC‹Pµ CЂP°P·CЂCЏPґC‹
PsCЃC,P°PIC€PëPjPëCЃCЏ CЃPëPjPIPsP»P°PjPë
    for i, num_at in enumerate(num):
        max_usable_at = gen_usable(len(num) - i, avail_chars, max)


        if max_usable_at and litval(max_usable_at) > litval(num_at):
            avail_chars[num_at] = avail_chars.get(num_at, 0) + 1
            num[i] = max_usable_at
            use(avail_chars, max_usable_at)


    if not num:
        print('-1')
        return


    print(''.join(num))
```

main()

Задание 4. Попытка 1.

```
room_count, wires_count = map(int, input().split())
room_conns: list[set[int]] = [set() for _ in range(room_count)]


for _ in range(wires_count):
    r1, r2 = map(int, input().split())
    r1, r2 = r1 - 1, r2 - 1
    if r1 != r2:
        room_conns[r1].add(r2)
        room_conns[r2].add(r1)


disconn_room = 0
for idx, conn_list in enumerate(room_conns):
    if len(conn_list) < len(room_conns[disconn_room]):
        disconn_room = idx


conn_pairs = [f"{min((disconn_room, conn)) + 1} {max((disconn_room, conn)) + 1}" for
conn in room_conns[disconn_room]]
print(len(room_conns[disconn_room]))
print('\n'.join(sorted(conn_pairs)))
```

Задание 4. Попытка 2.

```
room_count, wires_count = map(int, input().split())
room_conns: list[list[int]] = [[] for _ in range(room_count)]


for _ in range(wires_count):
    r1, r2 = map(int, input().split())
    r1, r2 = r1 - 1, r2 - 1
    if r1 != r2:
        room_conns[r1].append(r2)
        room_conns[r2].append(r1)


least_conn_idx = 0
for room_idx, room in enumerate(room_conns):
    if len(room) < len(room_conns[least_conn_idx]):
        least_conn_idx = room_idx


deleted_pairs = [f"{min(least_conn_idx, conn_idx) + 1} {max(least_conn_idx, conn_idx)
+ 1}"
                 for conn_idx in room_conns[least_conn_idx]]
print(len(deleted_pairs))
print('\n'.join(sorted(deleted_pairs)))
```