



**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. ЛОМОНОСОВА**

ОЛИМПИАДНАЯ РАБОТА

Наименование олимпиады школьников: **«Ломоносов»**

Профиль олимпиады: **Информатика**

ФИО участника олимпиады: **Камалов Вадим Ильдарович**

Класс: **9 класс**

Технический балл: **77**

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	64	86	0	72	0

Задание 1. Попытка 1.

```
k = int(input())
```

```
n = int(input())
```

```
x = []
```

```
for _ in range(n):
```

```
    x.append(input())
```

```
def fact(n):
```

```
    saved = {1: 1, 2: 2, 3: 6, 4: 24, 5: 120, 6: 720, 7: 5040, 8: 40320, 9: 362880, 10: 3628800, 11: 39916800, 12: 479001600, 13: 6227020800, 14: 87178291200, 15: 1307674368000, 16: 20922789888000, 17: 355687428096000, 18: 6402373705728000, 19: 121645100408832000, 20: 2432902008176640000, 21: 51090942171709440000, 22: 1124000727777607680000, 23: 25852016738884976640000, 24: 620448401733239439360000, 25: 15511210043330985984000000, 26: 403291461126605635584000000, 27: 10888869450418352160768000000, 28: 304888344611713860501504000000, 29: 8841761993739701954543616000000, 30: 26525285981219105863630848000000, 31: 8222838654177922817725562880000000, 32: 263130836933693530167218012160000000, 33: 868331761881188649551819440128000000, 34: 29523279903960414084761860964352000000, 35: 1033314796638614492966665133752320000000, 36: 37199332678990121746799944815083520000000, 37: 1376375309122634504631597958158090240000000, 38: 52302261746660111176000722410007429120000000, 39: 2039788208119744335864028173990289735680000000}
```

```
    if n <= 39:
```

```
        return saved[n]
```

```
    if n == 1:
```

```
        return 1
```

```
    return n * fact(n - 1)
```

```
k_fact = fact(k)
```

```
def val(d):
```

```
    if d.isdigit():
```

```
        return int(d)
```

```
    return ord(d) - 55
```

```
def to_number(x):
```

```
    res = 0
```

```
    to_cut = -1
```

```
    len_x = len(x)
```

```
    for i in range(len_x):
```

```
        if to_cut == -1 and x[i] != "0":
```

```
            to_cut = i
```

```
            continue
```

```
        res += val(x[i]) * fact(len_x - i)
```

```
    return (res, x[to_cut:])
```

```
found = False
```

```
for i in range(len(x) - 1, -1, -1):
```

```
    res, good = to_number(x[i])
```

```
if res % k_fact == 0:  
    print(good, i + 1, sep="\n")  
    found = True  
    break  
if not found:  
    print(-1)
```

Задание 1. Попытка 2.

```
def main():

    with open("input.txt", "r") as f:

        inp = f.read().split("\n")

        k = int(inp[0])

        n = int(inp[1])

        x = inp[2:]

    # k = int(input())

    # n = int(input())

    # x = []

    # for _ in range(n):

    #     x.append(input())

def fact(n):

    saved = {1: 1, 2: 2, 3: 6, 4: 24, 5: 120, 6: 720, 7: 5040, 8: 40320, 9: 362880, 10:
3628800, 11: 39916800, 12: 479001600, 13: 6227020800, 14: 87178291200, 15:
1307674368000, 16: 20922789888000, 17: 355687428096000, 18: 6402373705728000,
19: 121645100408832000, 20: 2432902008176640000, 21: 51090942171709440000, 22:
1124000727777607680000, 23: 25852016738884976640000, 24:
620448401733239439360000, 25: 15511210043330985984000000, 26:
403291461126605635584000000, 27: 10888869450418352160768000000, 28:
304888344611713860501504000000, 29: 8841761993739701954543616000000, 30:
26525285981219105863630848000000, 31: 8222838654177922817725562880000000,
32: 263130836933693530167218012160000000, 33:
8683317618811886495518194401280000000, 34:
295232799039604140847618609643520000000, 35:
10333147966386144929666651337523200000000}

    if n <= 39:

        return saved[n]

    if n == 1:
```

```
        return 1
    return n * fact(n - 1)

k_fact = fact(k)

def val(d):
    if d.isdigit():
        return int(d)

    return ord(d) - 55

def to_number(x):
    res = 0
    to_cut = -1
    len_x = len(x)
    for i in range(len_x):
        if to_cut == -1 and x[i] != "0":
            to_cut = i
            continue

    res += val(x[i]) * fact(len_x - i)

    return (res, x[to_cut:])

found = False
```

```
for i in range(len(x) - 1, -1, -1):
    res, good = to_number(x[i])
    if res % k_fact == 0:
        print(good, i + 1, sep="\n")
        found = True
        break
if not found:
    print(-1)

main()
```


Задание 1. Попытка 3.

```
def main():

    with open("input.txt", "r") as f:

        inp = f.read().split("\n")

        k = int(inp[0])

        n = int(inp[1])

        x = inp[2:]

    # k = int(input())

    # n = int(input())

    # x = []

    # for _ in range(n):

    #     x.append(input())

def fact(n):

    saved = {1: 1, 2: 2, 3: 6, 4: 24, 5: 120, 6: 720, 7: 5040, 8: 40320, 9: 362880, 10:
3628800, 11: 39916800, 12: 479001600, 13: 6227020800, 14: 87178291200, 15:
1307674368000, 16: 20922789888000, 17: 355687428096000, 18: 6402373705728000,
19: 121645100408832000, 20: 2432902008176640000, 21: 51090942171709440000, 22:
1124000727777607680000, 23: 25852016738884976640000, 24:
620448401733239439360000, 25: 15511210043330985984000000, 26:
403291461126605635584000000, 27: 10888869450418352160768000000, 28:
304888344611713860501504000000, 29: 8841761993739701954543616000000, 30:
26525285981219105863630848000000, 31: 8222838654177922817725562880000000,
32: 263130836933693530167218012160000000, 33:
8683317618811886495518194401280000000, 34:
295232799039604140847618609643520000000, 35:
10333147966386144929666651337523200000000}

    if n <= 35:

        return saved[n]

    if n == 1:
```

```
        return 1
    return n * fact(n - 1)

k_fact = fact(k)

def val(d):
    if d.isdigit():
        return int(d)

    return ord(d) - 55

def to_number(x):
    res = 0
    to_cut = -1
    len_x = len(x)
    for i in range(len_x):
        if to_cut == -1 and x[i] != "0":
            to_cut = i
            continue

        res += val(x[i]) * fact(len_x - i)

    return (res, x[to_cut:])

found = False
```

```
for i in range(len(x) - 1, -1, -1):
    res, good = to_number(x[i])
    if res % k_fact == 0:
        print(good, i + 1, sep="\n")
        found = True
        break
if not found:
    print(-1)

main()
```

Задание 2. Попытка 1.

with open('input.txt', 'r') as f:

```
inp = f.read().split("\n")
```

```
n = int(inp[0])
```

```
x = inp[1]
```

```
def val(d):
```

```
    if d.isdigit():
```

```
        return int(d)
```

```
    return ord(d) - 55
```

```
def to_val(d):
```

```
    if d <= 9:
```

```
        return str(d)
```

```
    return chr(d + 55)
```

```
def remove_zeros(x):
```

```
    to_cut = -1
```

```
    for i in range(len(x)):
```

```
        if to_cut == -1 and x[i] != "0":
```

```
            to_cut = i
```

```
            break
```

```
    return x[to_cut:]
```

```
values = { }
```

```
for i in x:
```

```
    if i.isdigit() or 65 <= ord(i) <= 90:
```

```
        values[val(i)] = values.get(val(i), 0) + 1
```

```
save_values = values.copy()
```

```
good = ""
```

```
pos = 0
```

```
max_length = 0
```

```
while True:
```

```
    found_char = False
```

```
    for i in range(pos + 1, -1, -1):
```

```
        if values.get(i, 0) > 0:
```

```
            found_char = True
```

```
            values[i] -= 1
```

```
            pos += 1
```

```
            break
```

```
if not found_char or pos > 35:
```

```
    max_length = min(35, pos)
```

```
break
```

```
values = save_values
```

```
result = ""
```

```
for pos in range(max_length, -1, -1):
```

```
    for i in range(pos + 1, -1, -1):
```

```
        if values.get(i, 0) > 0:
```

```
            found_char = True
```

```
            values[i] -= 1
```

```
            result += to_val(i)
```

```
            break
```

```
if len(result) > 35:
```

```
    result = result[:35]
```

```
if not result:
```

```
    print(-1)
```

```
else:
```

```
    print(remove_zeros(result))
```

Задание 3. Попытка 1.

```
x = int(input())
input()
face = [[] for _ in range(2 * (x + 1))]
while True:
    inp = input()
    if inp == "BACK":
        break
    inp = list(map(int, inp.split()))
    inp[1] = inp[1] + x
    face[inp[0]].append(inp[1])
    face[inp[1]].append(inp[0])
back = [[] for _ in range(2 * (x + 1))]
while True:
    inp = input()
    if inp == "END":
        break
    inp = list(map(int, inp.split()))
    inp[1] = inp[1] + x
    back[inp[0]].append(inp[1])
    back[inp[1]].append(inp[0])

visited = [False for _ in range(2 * (x + 1))]

def dfs(n, direction):
```

```

global visited
visited[n] = True
if direction == 1: # C%oP°CΓ PSCrP¶PSPs PëPrC,Pë PSP°PIPμCᄁC...Cf
    for i in face[n]:
        if not visited[i]:
            dfs(i, 0)
else: # PSCrP¶PSPs PëPrC,Pë PIPSPëP·
    for i in back[n]:
        if not visited[i]:
            dfs(i, 1)

dfs(1, 1)

good = True
for i in visited[1:-1]:
    if not i:
        good = False
        break

if good:
    print("YES")
else:
    print("NO")

```


Задание 3. Попытка 2.

```
x = int(input())
```

```
input()
```

```
face = [[] for _ in range(2 * (x + 1))]
```

```
used = set()
```

```
while True:
```

```
    inp = input()
```

```
    if inp == "BACK":
```

```
        break
```

```
    inp = list(map(int, inp.split()))
```

```
    inp[1] = inp[1] + x
```

```
    face[inp[0]].append(inp[1])
```

```
    face[inp[1]].append(inp[0])
```

```
    used.add(inp[0])
```

```
    used.add(inp[1])
```

```
back = [[] for _ in range(2 * (x + 1))]
```

```
while True:
```

```
    inp = input()
```

```
    if inp == "END":
```

```
        break
```

```
    inp = list(map(int, inp.split()))
```

```
    inp[1] = inp[1] + x
```

```
    back[inp[0]].append(inp[1])
```

```
back[inp[1]].append(inp[0])
```

```
used.add(inp[0])
```

```
used.add(inp[1])
```

```
visited = [False for _ in range(2 * (x + 1))]
```

```
def dfs(n, direction):
```

```
    global visited
```

```
    visited[n] = True
```

```
    if direction == 1: # C%oP°CÍ PSCrP¶PSPs PëPrC,Pë PSP°PIPμCḂC...Cf
```

```
        for i in face[n]:
```

```
            if not visited[i]:
```

```
                dfs(i, 0)
```

```
    else: # PSCrP¶PSPs PëPrC,Pë PIPSPëP·
```

```
        for i in back[n]:
```

```
            if not visited[i]:
```

```
                dfs(i, 1)
```

```
dfs(1, 1)
```

```
used = list(used)
```

```
good = True
```

```
for i in used:
```

```
    if not visited[i]:
```

```
good = False
```

```
break
```

```
if not good:
```

```
    dfs(1, 0)
```

```
    good = True
```

```
    visited = [False for _ in range(2 * (x + 1))]
```

```
    for i in used:
```

```
        if not visited[i]:
```

```
            good = False
```

```
            break
```

```
if good:
```

```
    print("YES")
```

```
else:
```

```
    print("NO")
```

Задание 4. Попытка 1.

```
n, m = map(int, input().split())
p = [[] for _ in range(n)]
for _ in range(m):
    inp = list(map(int, input().split()))
    inp[0] -= 1
    inp[1] -= 1
    p[inp[0]].append(inp[1])
    p[inp[1]].append(inp[0])

def dfs(v):
    visited[v] = True
    for i in p[v]:
        if not visited[i]:
            dfs(i)

candidate = (-1, 1e18)

for vertex in range(n):
    if len(p[vertex]) < candidate[1]:
        candidate = (vertex, len(p[vertex]))
    # visited = [False for _ in range(n)]
    # visited[vertex] = True
    # to_go = 0 if vertex != 0 else 1
    # dfs(to_go)
```

```
# if visited.count()

to_remove = []
for i in p[candidate[0]]:
    line = [min(candidate[0], i) + 1, max(candidate[0], i) + 1]
    line = str(line[0]) + " " + str(line[1])
    to_remove.append(line)

print(candidate[1])
print(*sorted(to_remove), sep="\n")
```

Задание 4. Попытка 2.

```
from sys import exit

n, m = map(int, input().split())

p = [[] for _ in range(n)]

for _ in range(m):

    inp = list(map(int, input().split()))

    inp[0] -= 1

    inp[1] -= 1

    if inp[0] != inp[1]:

        p[inp[0]].append(inp[1])

        p[inp[1]].append(inp[0])

def dfs(v):

    free[v] = False

    for i in p[v]:

        if free[i]:

            dfs(i)

free = [True for _ in range(n)]

dfs(0)

if any(free):

    print(0)

    exit()

candidate = (-1, 1e18)
```

```
for vertex in range(n):
    if len(p[vertex]) < candidate[1]:
        candidate = (vertex, len(p[vertex]))
    # visited = [False for _ in range(n)]
    # visited[vertex] = True
    # to_go = 0 if vertex != 0 else 1
    # dfs(to_go)
    # if visited.count()

to_remove = []
for i in p[candidate[0]]:
    line = list(sorted([candidate[0] + 1, i + 1]))
    line = str(line[0]) + " " + str(line[1])
    to_remove.append(line)

print(candidate[1])
print(*sorted(to_remove), sep="\n")
```

Задание 5. Попытка 1.

```
a1, b1, a2, b2 = map(int, input().split())
```

```
start = (a1, b1)
```

```
end = (a2, b2)
```

```
if start == end:
```

```
    print(1)
```

```
else:
```

```
    print((a1 - a2) ** 2 + (b1 - b2) ** 2)
```


Задание 5. Попытка 2.

```
a1, b1, a2, b2 = map(int, input().split())
```

```
start = (a1, b1)
```

```
end = (a2, b2)
```

```
if start == end:
```

```
    print(1)
```

```
elif a1 == a2:
```

```
    print(abs(b1 - b2) + 1)
```

```
else:
```

```
    ans = (a1 - a2) ** 2 + (b1 - b2) ** 2
```

```
    print(max(2, ans))
```