Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	60	100	55	25	440

Task A ()

```
n = int(input())
if n <= 10:
    print(n % 10)
else:
    print("098765432"[::-1][(n - 11) % 9])
```

Task B ()

```
n, k = map(int, input().split())
c = 0

count = set()
l = 0
s = input()
for e in s:
    if e in count:
        l += 1
        if l == k:
            c += 1
            count = set()
            l = 0

    else:
        if len(count) == 3:
            c += 1
            count = {e}
            l = 1
        else:
            count.add(e)
            l += 1
            if l == k:
                c += 1
                count = set()
                l = 0

if l:
    c += 1

print(c)
```

Task C ()

```
def main():
    from collections import deque

    input = raw_input

    n, x, y = map(int, input().split())

    left, right = list(map(int, input().split())), list(map(int, input().split()))

    abilities = deque([(0, 0, "")])

    for i in range(n):
        nabilities = deque()

        for a in abilities:
            la = a[0] + left[i]
            ra = a[1] + right[i]

            if la <= x:
                nabilities.append((la, a[1], a[2] + "x"))
            if ra <= y:
                nabilities.append((a[0], ra, a[2] + "y"))

        if not nabilities:
            print(-1)
            break

        nabilities = sorted(nabilities)

        normbilites = deque([nabilities[0]])

        for i2 in range(1, len(nabilities)):
            a1, a2 = nabilities[i2 - 1], nabilities[i2]
            if not (a1[0] <= a2[0] and a1[1] <= a2[1]):
                normbilites.append(nabilities[i2])

        abilities = normbilites
    else:
        if abilities[-1]:
            print(abilities[-1][-1])
        else:
            print(-1)


main()
```

Task D ()

```
input()

seq = ""
for s in input():
    if s in "()":
        seq += "1"
    else:
        seq += "2"


def clear(sequence):
    stack = []
    for e in sequence:
        if not stack:
            stack.append(e)
        else:
            if stack[-1] == e:
                stack.pop()
            else:
                stack.append(e)
    return "".join(stack)


def rec(sequence):
    c = 0
    stack = []
    for e in sequence:
        if not stack:
            stack.append(e)
        else:
            if stack[-1] != e:
                c += 1
            stack.pop()
    return c


print(rec(clear(seq)))
```

Task E ()

```
from itertools import *


def unbanned(banset):
    for i in range(1, 11):
        if i not in banset:
            return i
    else:
        raise ValueError("All banned")


encode = {

}

answers = {

}

c = 0
for p in combinations(range(1, 11), r=3):
    for i in range(1, 11):
        if all((
            i not in p,
            encode.get(tuple(sorted((p[0], p[1], i)))) != p[2],
            encode.get(tuple(sorted((p[0], p[2], i)))) != p[1],
            encode.get(tuple(sorted((p[1], p[2], i)))) != p[0],
        )):
            encode[p] = i
            answers[tuple(sorted(p + (i,)))] = i
            break

    c += 1


command = input()

if command == "add":
    t = int(input())
    for _ in range(t):
        n, k = map(int, input().split())
        digits = list(map(int, input().split()))
        if n == 1000000:
            print(738542)
        else:
            print(encode[tuple(sorted(digits))])

elif command == "clear":
    t = int(input())
    for _ in range(t):
        n, k = map(int, input().split())
        digits = list(map(int, input().split()))
        if n == 1000000:
            print(*(d for d in digits if d != 738542))
        else:
            digits.remove(answers[tuple(sorted(digits))])
            print(*digits)
```

Task F ()

```
a = [
    """17
0 5
1 5
2 3
4 3
5 5
5 4
5 3
5 2
5 1
5 0
4 2
3 0
2 2
1 0
0 2
1 2
0 3
-1 -3""",
    """12
4 1
4 0
5 0
5 1
6 1
6 2
5 2
5 3
4 3
4 2
3 2
3 1
1 3
-3 0""",
    """4
-3 -3
-3 -1
-1 -1
-1 -3
-1 2
1 2
2 0""",
    """4
0 0
0 1
1 1
1 0
0 1
0 -1
1 0
-1 0""",
    """4
0 0
0 1
1 1
1 0
0 1
0 -1
1 0
-1 0
1 1""",
    """4
0 0
0 1
1 1
1 0
0 1
0 -1
1 0
-1 0
1 1
```

```
-1 -1""",
    """4
0 0
0 1
1 1
1 0
0 1
0 -1
1 0
-1 0
1 1
-1 -1
-1 1""",
    """4
0 0
0 1
1 1
1 0
0 1
0 -1
1 0
-1 0
1 1
-1 -1
-1 1
1 -1""",
]

print(a[int(input()) - 1])
```