

Олимпиада СПбГУ по информатике 2021/22 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	58	0	458

Task A ()

```
#include <iostream>
#include <vector>
#include <map>
#include <utility>
#include <algorithm>

using namespace std;

int main() {
    int n;
    cin >> n;
    map<int, int> num;
    for (int i = 0; i < n; ++i) {
        int a, b;
        cin >> a >> b;
        int j = 0;
        while(a != 0) {
            num[j + b] += a % 10;
            if (num[j + b] >= 10) {
                int k = 0;
                while (num[j + b + k] >= 10) {
                    num[j + b + k + 1] += 1;
                    num[j + b + k] -= 10;
                    k += 1;
                }
            }
            a /= 10;
            j += 1;
        }
    }
    map<int, int> num1 = num;

    vector<pair<int, int>> rem;
    for (auto i: num1)
        rem.push_back(i);
    sort(rem.begin(), rem.end());
    for (auto i: rem) {
        if (num1[i.first] >= 10) {
            num1[i.first] -= 10;
            num1[i.first + 1]++;
        }
    }
    int min_n0 = 1000000100;
    for (auto i: num1) {
        if (i.second != 0)
            min_n0 = min(min_n0, i.first);
    }
    cout << min_n0;
}
```

Task B ()

```
n = int(input())
a = list(map(int, input().split()))
b = 0
for i in range(n):
    if a[i] != 0:
        b = i
t = dict()
print('Flip_and_wait', flush=True)
i = 0
am = 0
while True:
    i += 1
    s = input()
    up = s.count("e") // 2
    t[i] = [up, sum(a[am: am + up])]
    am += up
    if am - 1 >= b:
        break
    print('Wait', flush=True)
newt = []
t[i][1] -= 1
for i in t:
    newt.append([i, t[i]])
newt.sort()
newt.reverse()
for i in range(len(newt)):
    for j in range(newt[i][1][1]):
        print('Flip_and_wait', flush=True)
        s = input()
        for k in range(len(newt) - i - 1):
            print('Wait', flush=True)
            s = input()
print('Stop', flush=True)
```

Task C ()

```
def remake2(a):
    s = ''
    if a.count("1") >= len(a) // 2:
        for i in a:
            if i == "1":
                s += '?'
            else:
                s += '0'
    else:
        for i in a:
            if i == "0":
                s += '?'
            else:
                s += '1'
    return s

s1 = [0, 0, '?1', '??1', '??11', '???11', '???111', '???1111', '???11111', '???111111', '???1111111']
t = int(input())
for i in range(t):
    s = input()
    if s.count("?"):
        if s.count("?") == len(s):
            print('0' * len(s))
        elif s == (len(s) - len(s) // 2) * '?' + '1' * (len(s) // 2):
            print('1' * len(s))
        else:
            if s.count("0"):
                for i in s:
                    if i == '?':
                        print('1', end='')
                    else:
                        print('0', end='')
            else:
                for i in s:
                    if i == '?':
                        print('0', end='')
                    else:
                        print('1', end='')
            print()
    else:
        if s.count("1") == len(s):
            print((len(s) - len(s) // 2) * '?' + '1' * (len(s) // 2))
        else:
            print(remake2(s))
```

Task D ()

```
#include <iostream>
#include <vector>
#include <map>
#include <utility>
#include <algorithm>

using namespace std;

const long long MOD = 998244353;
vector<int> pow = {1, 295201906, 160030060, 957629942, 545208507, 213689172, 760025067, 939830261,
    506268060, 39806322, 808258749, 440133909, 686156489, 741797144, 390377694, 12629586,
    544711799, 104121967, 495867250, 421290700, 117153405, 57084755, 202713771, 675932866,
    79781699, 956276337, 652678397, 35212756, 655645460, 468129309, 761699708, 533047427,
    287671032, 206068022, 50865043, 144980423, 111276893, 259415897, 444094191, 593907889,
    573994984, 892454686, 566073550, 128761001, 888483202, 251718753, 548033568, 428105027,
    742756734, 546182474, 62402409, 102052166, 826426395, 159186619, 926316039, 176055335,
    51568171, 414163604, 604947226, 681666415, 511621808, 924112080, 265769800, 955559118,
    763148293, 472709375, 19536133, 860830935, 290471030, 851685235, 242726978, 169855231,
    612759169, 599797734, 961628039, 953297493, 62806842, 37844313, 909741023, 689361523,
    887890124, 380694152, 669317759, 367270918, 806951470, 843736533, 377403437, 945260111,
    786127243, 80918046, 875880304, 364983542, 623250998, 598764068, 804930040, 24257676,
    214821357, 791011898, 954947696, 183092975};
const long long prevp = 998244352;

long long calc_nfact(long long n) {
    long long start_p = pow[n / 10000000];
    for (int i = n / 10000000 * 10000000 + 1; i <= n; ++i) {
        start_p = (start_p * i) % MOD;
    }
    return start_p;
}

long long calc_nfact_ez(long long n) {
    long long start_p = 1;
    for (int i = 2; i <= n; ++i) {
        start_p = (start_p * i) % MOD;
    }
    return start_p;
}

int main() {
    long long n;
    cin >> n;
    n += 1;
    long long am = 0;
    am = n / MOD + n / (MOD * MOD);
    long long pown;
    if ((n / MOD) % 2 == 0)
        pown = 1;
    else
        pown = prevp;
    cout << am << "_" << (pown * calc_nfact(n % MOD)) % MOD << "\n";
}
```

Task E ()

```
#include <iostream>
#include <vector>
#include <map>
#include <utility>
#include <algorithm>

#pragma optimize("Ofast")

using namespace std;

int main() {
    ios_base::sync_with_stdio(false);
    cout.tie(0);
    int n;
    cin >> n;
    vector<int> a(n);
    for (int i = 0; i < n; ++i) {
        cin >> a[i];
    }
    int q;
    cin >> q;
    for (int i = 0; i < q; ++i) {
        int a1, b, q1;
        cin >> a1 >> b >> q1;
        long long ans = 0;
        for (int j = a1; j < b; ++j) {
            ans += (q1 + a[j] - 1) / a[j];
        }
        cout << ans << "\n";
    }
}
```

Task F ()