

Задача А. Дойди до нуля

Если выбранное наугад число – 1, то Петя своим первым ходом не может выбрать четного числа, чтобы вычесть его из исходного. В этом случае Петя проиграет. Если выбранное число, в общем случае, нечетное, то Петя первым ходом может выбрать любое четное число и вычесть из n . При вычитании из нечетного числа четного получаем нечетное $n=n-a$. И Вася своим вторым ходом выберет число n , произведет вычитание и получит число 0. Петя снова проиграет. Таким образом, если n – нечетное, то проигрывает Петя, выигрывает Вася. Нужно вывести число 2 в ответе.

Если задуманное число – четное, то в этом случае Петя сразу выберет число, равное исходному, вычтет его из исходного и выиграет своим первым ходом. Таким образом, если n – четное, то выиграет при оптимальной игре Петя. Нужно вывести число 1.

Приведем фрагмент программы решения задачи.

```
int n;  
scanf("%d",&n);  
if (n%2)  
    printf(2);  
else  
    printf(1);
```

Задача В. Строим параллелограммы

Произведем подсчет количества палочек каждой длины. Для этого будем использовать массив `cnt[]`. Для вычисления ответа заметим, что в параллелограмм входят две пары равных сторон. Поэтому сначала вычислим количество пар равных по длине сторон $ans += cnt[i] / 2$. В дальнейшем из каждой двух пар палочек получится параллелограмм. Поэтому ответом к задаче будет $ans / 2$.

```
int n;  
scanf("%d", &n);  
for (int i = 0; i < n; ++i) {  
    int cur;  
    scanf("%d", &cur);
```

```
        ++cnt[cur];
    }
    int ans = 0;
    for (int i = 0; i < N; ++i) {
        ans += cnt[i] / 2;
    }
    printf("%d\n", ans / 2);
```

Задача С. Распродажа

Отрицательные элементы складываем в массив, сортируем, находим сумму m наибольших по модулю. Формально решение выглядит следующим образом: отрицательные элементы складываем в массив, сортируем, находим сумму m наибольших по модулю. Отрицательные элементы – это деньги, которые Дима может заработать. По принципу жадности «покупаем» сначала те телефоны, которые дадут наибольшую сумму заработка, но не забываем, что Дима может унести не более m телефонов.

```
sort(a, a + n);
long long ans = 0;
for (int i = 0; i < m; ++i) {
    if (a[i] >= 0) {
        break;
    }
    ans -= a[i];
}
```

Задача D. Игра

Для каждой коробки в массиве $wh[i]$ будем хранить номер стопки, в которой она находится. Изначально каждая коробка представляет собой стопку, состоящую из ее самой. Для каждой коробки – основания стопки будем использовать массив, элементами которого будут коробки,

находящиеся в этой стопке. Рассмотрим фрагмент псевдокода для решения задачи.

```
vector<int> a[N];
int p[N];
...
for (int i = 0; i < n; ++i) {
    p[i] = i;
    a[i].push_back(i);
}
while (q-- > 0) {
    int type;
    cin >> type;
    if (type == 1) {
        int x, y;
        cin >> x >> y;
        a[p[y - 1]].push_back(x - 1);
        a[p[x - 1]].pop_back();
        p[x - 1] = p[y - 1];
    } else {
        int x;
        cin >> x;
        cout << a[p[x - 1]].back() + 1 << '\n';
    }
}
```