

| | |
|---------------------------|-------------------------------------|
| Тест начат | Понедельник, 24 октября 2022, 20:09 |
| Состояние | Завершено |
| Завершен | Понедельник, 24 октября 2022, 20:54 |
| Прошло времени | 45 мин. 17 сек. |
| Баллы | 6,00/8,00 |
| Оценка | 7,50 из 10,00 (75%) |

Вопрос **1**

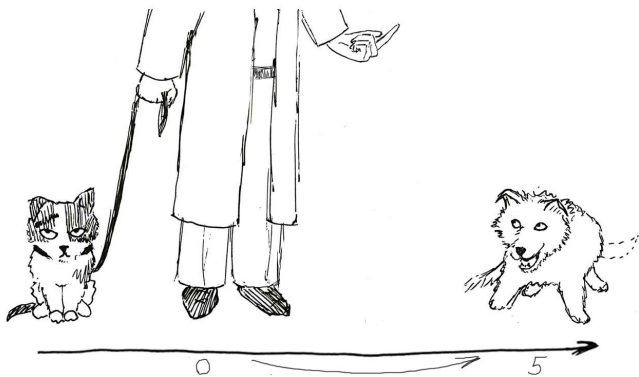
Верно

Баллов: 1,00 из 1,00

Белка

Белка сорвалась с поводка и прыгает по узкому коридору вокруг Королёва. В начальный момент она находится рядом с ним в координате 0. Прыжки с четным индексом она делает в положительном направлении, с нечетным индексом - в отрицательном. Нумерация прыжков - с нуля.

Рассчитайте, куда попадет Белка после нескольких прыжков.



Входные данные:

Натуральное число n ($1 \leq n \leq 100$).

Последовательность из n чисел - длины прыжков (тоже в диапазоне от 1 до 100).

Выходные данные:

Одно целое число - координата Белки после всех прыжков.

Например:

| Ввод | Результат |
|----------------|-----------|
| 5 1 1 1 1 1 | 1 |
| 1 5 | 5 |

Ответ: (штрафной режим: 0, 0, 0, 10, 20, ...%)

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(){
6     int n, s, x; cin >> n;
7     s = 0;
8     for (int i = 0; i < n; i++){ // прыгаем n раз
9         cin >> x;
10        if (i % 2 == 0) s += x; // проверка на четность прыжка, если
11        else s -= x; // иначе влево
12    }
13    cout << s;
14    return 0;
15 }
16
```

| | Ввод | Ожидается | Получил | |
|---|--|-----------|---------|---|
| ✓ | 5 1 1 1 1 1 | 1 | 1 | ✓ |
| ✓ | 1 5 | 5 | 5 | ✓ |
| ✓ | 5 1 1 1 2 1 | 0 | 0 | ✓ |
| ✓ | 5 10 5 1 5 1 | 2 | 2 | ✓ |
| ✓ | 2 10 10 | 0 | 0 | ✓ |
| ✓ | 2 1 2 | -1 | -1 | ✓ |
| ✓ | 100 2 3 4 5 2 1 20 46 1 15 49 31 54 1 1 33 4 5 4 8 2 3 4 5 2 1 20 46 1 15 49 31 54 1 1 33 4 5 4 8 2 3 4 5 2 1 20 46 1 15 49 31 54 1 1 33 4 5 4 8 2 3 4 5 2 1 20 46 1 15 49 31 54 1 1 33 4 5 4 8 2 3 4 5 2 1 20 46 1 15 49 31 54 1 1 33 4 5 4 8 | -35 | -35 | ✓ |

| | | | | |
|--|-------------------------|------------------|----------------|--|
| | 49 31 54 1 1 33 4 5 4 8 | | | |
| | Ввод | Ожидается | Получил | |

| | | | | |
|---|--------------------------------------|------|------|---|
| ✓ | 10 95 32 5 46 4 32 4 698 4524 354 | 3470 | 3470 | ✓ |
|---|--------------------------------------|------|------|---|

Все тесты пройдены! ✓

Решение автора вопроса:

```
#include <iostream>
using namespace std;

int main()
{
    int n, res = 0, cur;
    cin >> n;
    for (int i = 0; i < n; i++)
    {
        cin >> cur;
        if (i % 2)
            res -= cur;
        else
            res += cur;
    }
    cout << res << endl;
    return 0;
}
```

Верно
Баллы за эту попытку: 1,00/1,00.

Вопрос **2**

Верно

Баллов: 1,00 из 1,00

В поисках неона или молибдена

Вы вешали на стену таблицу Менделеева, приколачивая по одной табличке с элементом за раз. Начали, разумеется, снизу - с последних элементов. Потом вам понадобилась лестница, и вы ушли за ней в подсобку. А там пока чай попить, пока что... Когда вы вернулись, кот уже успел разбросать таблички по всей комнате. Одну он утащил на шкаф и там грызет. На попытки отобрать табличку он очень страшно шипит.

Кое-как вы собрали таблички в одну стопку. Теперь вам нужно понять, какую из них надо печатать заново. Вы не хотите связываться с этим котом.

Дана неотсортированная последовательность, которая содержит все натуральные числа от 1 до N, кроме одного числа.

Нужно определить, какого числа не хватает.

Входные данные:

Натуральное число N ($2 \leq N \leq 118$).

Последовательность из N - 1 чисел.

Выходные данные:

Одно целое число - недостающий элемент.

Например:

| Ввод | Результат |
|--------------|-----------|
| 5 4 5 2 1 | 3 |
| 2 2 | 1 |

Ответ: (штрафной режим: 0, 0, 0, 10, 20, ...%)

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(){
6     int N, m[118], n[118], j = 0; cin >> N;
7     for (int i = 1; i <= N; i++){ // заполняем "эталонный" массив,
8         n[j] = i; j++;
9     }
10    for (int i = 0; i < N - 1; i++) cin >> m[i]; // заполняем массив
11    for (int i = 0; i < N - 1; i++){ // проводим сортировку пузырьком
12        for (int j = 0; j < N - 2 - i; j++){
13            if (m[j] > m[j + 1]){
14                int tmp = m[j];
15                m[j] = m[j + 1];
16                m[j + 1] = tmp;
17            }
18        }
19    }
20    for (int i = 0; i < N; i++){ // ищем первое несоответствие между
21        if (m[i] != n[i]){ // когда это произойдет, i-е число эталон
22    }
```

| | Ввод | Ожидается | Получил | |
|---|--|-----------|---------|---|
| ✓ | 5 4 5 2 1 | 3 | 3 | ✓ |
| ✓ | 2 2 | 1 | 1 | ✓ |
| ✓ | 10 9 7 6 5 4 1 3 2 10 | 8 | 8 | ✓ |
| ✓ | 118 118 117 116 115 114 113 112 111 110 1 2 3 4 5 6 7 8 9 10 30 29 28 27 26 25 24 23 22 21 20 11 12 13 14 15 16 17 18 19 100 101 102 103 104 105 106 107 108 109 31 32 33 34 35 36 37 38 39 40 41 51 80 52 43 53 44 54 45 55 46 56 47 57 48 58 49 59 50 60 61 62 63 64 65 66 67 68 69 70 99 98 97 96 95 94 93 92 91 90 71 89 72 88 73 87 74 86 75 85 76 84 77 83 78 82 79 81 | 42 | 42 | ✓ |
| ✓ | 118 118 117 116 115 114 113 112 111 110 1 2 3 4 5 6 7 8 9 42 30 29 28 27 26 25 24 23 22 21 20 11 12 13 14 15 16 17 18 19 100 101 102 103 104 105 106 107 108 109 31 32 33 34 35 36 37 38 39 40 41 51 80 52 43 53 44 54 45 55 46 56 47 57 48 58 49 59 50 60 61 62 63 64 65 66 67 68 69 70 99 98 97 96 95 94 93 92 91 90 71 89 72 88 73 87 74 86 75 85 76 84 77 83 78 82 79 81 | 10 | 10 | ✓ |

Все тесты пройдены! ✓

Решение автора вопроса:

```
#include <iostream>
using namespace std;

int main()
{
    int n, a[128];
    cin >> n;
    n--;
    for (int i = 0; i < n; i++)
        cin >> a[i];
    for (int i = 0; i < n; i++)
        for (int j = 0; j < n - i - 1; j++)
            if (a[j] > a[j + 1])
            {
                int tmp = a[j];
                a[j] = a[j + 1];
                a[j + 1] = tmp;
            }
    //for (int i = 0; i < n; i++)
    //    cout << a[i] << " ";
    //cout << endl;
    for (int i = 0; i < n; i++)
        if (i + 1 != a[i])
        {
            cout << i + 1 << endl;
            break;
        }
    return 0;
}
```

Верно

Баллы за эту попытку: 1,00/1,00.

Вопрос **3**
Верно
Баллов: 1,00 из 1,00

Праймориал

Напишите функцию, которая получает натуральное число и считает его праймориал.

```
unsigned long long int primorial(int n);
```

Праймориал числа - это произведение всех простых чисел, которые меньше его или ему равны. Как факториал, но перемножаем только простые множители.

Праймориалы это довольно забавная точка входа в теорию чисел. Посмотрите сами - если вы возьмете какой-то праймориал и добавите к нему единицу, то вы получите простое число. Из этого очевидно следует, что простых чисел на самом деле бесконечное множество. Если еще немного покрутить конструкцию, то с ее помощью можно искать арифметические прогрессии из простых чисел. Зачем это нужно - тут уже вы закопаетесь в криптографию и прочие интересные места.

Аргумент:

Одно натуральное число ($5 \leq n \leq 50$).

Возвращаемое значение:

Одно натуральное число - праймориал аргумента.

Инфраструктура (код, который можно использовать при отладке):

```
#include <iostream>
using std::cout;
using std::endl;
using std::cin;
unsigned long long int primorial(int n);
// ваша реализация функции
int main()
{
    int n;
    cin >> n;
    cout << primorial(n) << endl;
    return 0;
}
```

Внимание! Сдавать нужно только реализацию функции `primorial` и те функции, которые вы напишете для себя (например, функцию проверки на четность или делимость). Функцию `main` сдавать не нужно. Выводить что-либо на экран не нужно. Спасибо за внимание.

Например:

| Ввод | Результат |
|------|-----------|
| 5 | 30 |

Ответ: (штрафной режим: 0, 0, 0, 10, 20, ...%)

```
1 bool checker(int n){ // обыкновенная проверка на простоту числа
2     for (int i = 2; i < n; i++){ // не до корня из n, так как нет нуж
3         if (n % i == 0)
4             return false;
5     }
6     return true; // если число простое, возвращаем true, иначе false
7 }
8 unsigned long long int primorial(int n){
9     unsigned long long int p = 1; // наше начальное значение произвед
10    for (int i = 1; i <= n; i++){ // идем по числам от 1 до n включит
11        if (checker(i)){ // если число i простое, то домножаем на нег
12            p *= i;
13        }
14    }
15    return p; // возвращаем primorial
16 }
```

| | Ввод | Ожидается | Получил | |
|---|------|--------------------|--------------------|---|
| ✓ | 5 | 30 | 30 | ✓ |
| ✓ | 13 | 30030 | 30030 | ✓ |
| ✓ | 50 | 614889782588491410 | 614889782588491410 | ✓ |
| ✓ | 42 | 304250263527210 | 304250263527210 | ✓ |
| ✓ | 29 | 6469693230 | 6469693230 | ✓ |

Все тесты пройдены! ✓

Решение автора вопроса:

```
bool is_simple(int n)
{
    for (int i = 2; i <= n / 2; i++)
        if (n % i)
            return false;

    return true;
}

unsigned long long int primorial(int n)
{
    unsigned long long int res = 1;
    for (int i = 2; i <= n; i++)
    {
        if (is_simple(i))
            res *= i;
    }
    return res;
}
```

Верно

Баллы за эту попытку: 1,00/1,00.

Вопрос **4**

Верно

Баллов: 1,00 из 1,00

Ouija

Вы с друзьями хотели призвать призрак отца-основателя и спросить у него, какие билеты попадутся вам на экзамене. Но не стоило делать этого в октябре - что-то пошло не так, и пришел кто-то другой. Или другие. Они не хотят отвечать на вопросы, бормочут тарабарщину и норовят испортить доску, а то и сбросить ее со стола. Но делать нечего, придется все-таки расшифровывать их сообщения.

Напишите функцию, которая выводит на экран символы из глобального двумерного массива по заданному алгоритму.

```
char ouija[3][13] = {
    {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm'},
    {'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'},
    {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '.', '!', '?'}
};
```

```
void message(int i_start, int j_start, int length);
```

Первым выводится символ `ouija[i_start][j_start]`. Для того, чтобы находить следующие символы, нужно считывать с экрана по два числа, они будут обозначать смещение для первого индекса и для второго соответственно.

Если в какой-то момент смещения выводят вас за пределы массива, то нужно вывести строку "THE SPIRITS ARE RESTLESS". Других символов, которые вы уже успели считать, в этом случае выводить не надо.

- Аргументы:**
- Три целых числа:
- `i_start` - начальный индекс строки ($0 \leq i_start < 3$);
 - `j_start` - начальный индекс столбца ($0 \leq j_start < 13$);
 - `length` - длина сообщения ($1 \leq length \leq 50$).

Возвращаемое значение:

Отсутствует.

Инфраструктура (код, который можно использовать при отладке):

Перед тем, как вызвать вашу функцию, функция `main` "портит" таблицу `ouija`, меняя два символа местами.

```
#include <iostream>
using std::cout;
using std::endl;
using std::cin;

char ouija[3][13] = {
    {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm'},
    {'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'},
    {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '.', '!', '?'}
};

void message(int i_start, int j_start, int length);
// ваша реализация функции
```

```
int main()
{
    int i_swap_0, j_swap_0, i_swap_1, j_swap_1;
    cin >> i_swap_0 >> j_swap_0 >> i_swap_1 >> j_swap_1;
    char tmp = ouija[i_swap_0][j_swap_0];
    ouija[i_swap_0][j_swap_0] = ouija[i_swap_1][j_swap_1];
    ouija[i_swap_1][j_swap_1] = tmp;

    int i_start, j_start, length;
    cin >> i_start >> j_start >> length;
    message(i_start, j_start, length);

    return 0;
}
```

Внимание! Сдавать нужно только реализацию функции. Функцию `main` сдавать не нужно.

Выводить что-либо на экран **нужно**. Спасибо за внимание.

Например:

| Ввод | Результат |
|--|--------------------------|
| 0 0 0 1 0 0 5 0 1 0 1 0 1 0 1 | bacde |
| 0 0 0 1 0 0 5 0 1 0 -1 0 1 0 -1 | babab |
| 0 0 0 1 0 0 3 0 1 -5 -5 | THE SPIRITS ARE RESTLESS |

Ответ: (штрафной режим: 0, 0, 0, 10, 20, ...%)

```
1 void message(int i_start, int j_start, int length){
2     char s[50];
3     s[0] = ouija[i_start][j_start]; // начало нам уже дали, добавляем
4     int i_shift, j_shift; // инициализируем переменные перехода по i
5     for (int k = 1; k <= length - 1; k++){ // повторяем цикл length -
6         cin >> i_shift >> j_shift;
7         i_start += i_shift; j_start += j_shift; // используем старые
8         if (i_start < 0 || i_start >= 3 || j_start < 0 || j_start >=
9             cout << "THE SPIRITS ARE RESTLESS"; exit(0);
10        }
11        s[k] = ouija[i_start][j_start]; // иначе добавляем символ в с
12    }
13    for (int i = 0; i < length; i++) cout << s[i]; // выводим наш спи
14 }
```

| | Ввод | Ожидается | Получил | |
|---|---|--------------------------|--------------------------|---|
| ✓ | 0 0 0 1 0 0 5 0 1 0 1 0 1 0 1 | bacde | bacde | ✓ |
| ✓ | 0 0 0 1 0 0 5 0 1 0 -1 0 1 0 -1 | babab | babab | ✓ |
| ✓ | 0 0 0 1 0 0 3 0 1 -5 -5 | THE SPIRITS ARE RESTLESS | THE SPIRITS ARE RESTLESS | ✓ |
| ✓ | 0 1 2 10 2 10 22 -1 -9 0 10 0 -6 -1 -5 1 0 -1 3 0 3 0 2 1 -4 -1 7 1 -6 0 -4 -1 4 0 -1 1 4 -1 -4 1 0 0 7 -1 -11 0 6 0 -2 | boysandgirlsofeveryage | boysandgirlsofeveryage | ✓ |
| ✓ | 0 1 2 10 0 1 1 | . | . | ✓ |
| ✓ | 0 10 0 0 1 5 20 0 -4 -1 11 0 -8 1 2 -1 1 0 1 1 -8 -1 6 1 -1 0 2 0 -5 0 7 -1 -1 1 -2 -1 1 0 2 0 1 0 -8 0 -2 | somethingsupwithjack | somethingsupwithjack | ✓ |

| | Ввод | Ожидается | Получил | |
|---|---|------------------------------------|------------------------------------|---|
| ✓ | 0 4 0 8 0 4 34 1 -4 0 6 -1 1 0 1 0 -5 0 -3 1 4 -1 6 1 -9 -1 4 1 1 -1 1 0 1 1 -8 -1 4 0 2 0 1 1 -1 -1 3 1 -2 0 -2 0 1 -1 -5 0 7 0 -3 1 -4 0 3 -1 4 0 -5 0 -3 1 9 0 -9 1 11 | inthedarkofthenightjustbeforedawn! | inthedarkofthenightjustbeforedawn! | ✓ |

Все тесты пройдены! ✓

Решение автора вопроса:

```

void message(int i_start, int j_start, int length)
{
    char m[100];
    for (int l = 0; l < length; l++)
    {
        m[l] = ouija[i_start][j_start];
        if (l == length - 1) break;
        int i, j;
        cin >> i >> j;
        i_start += i;
        j_start += j;
        if (i_start >= 3 || j_start >= 13 || i_start < 0 || j_start < 0)
        {
            cout << "THE SPIRITS ARE RESTLESS" << endl;
            return;
        }
    }
    m[length] = '\0';
    cout << m << endl;
}

```

Верно
Баллы за эту попытку: 1,00/1,00.

Вопрос **5**

Верно

Баллов: 1,00 из 1,00

Космозо

В Московском космическом зоопарке проводят реконструкцию одной секции. Вольеры решили упорядочить в соответствии с маркетинговыми исследованиями и более популярные экспонаты поставить подальше - чтобы посетители прошли и мимо всех остальных тоже. Вам нужно написать функцию, которая поможет сравнить два экспоната в соответствии с общественными запросами. Не удивляйтесь, что получатся странные формулы - это эвристика, которая хоть как-то объясняет полученную при опросе статистику. Люди странные.

Напишите функцию, которая сравнивает две структуры.

```
struct AlienLifeForm
{
    char name[50];
    unsigned int id;
    unsigned int weight;
    unsigned int length;
    unsigned int speed;
    unsigned int tentacles;
    unsigned int color[4];
    bool sentient;
    bool carnivore;
};

int compare (AlienLifeForm a, AlienLifeForm b);
```

Большей является структура, для которой больше значение вау-фактора. Вау-фактор высчитывается по следующим правилам.

Если выполнено условие разумности (sentient == true), то вау-фактор получается сложением следующих значений:

- количество символов в name, умноженное на коэффициент 126; *имя длинное - наверное, умный*
- length, умноженный на коэффициент 5;
- tentacles, умноженный на коэффициент 97; *ктулха!*
- элементы массива color с индексами 0 и 1, умноженные на коэффициент 4;
- элемент массива color с индексом 2, умноженный на коэффициент 30; *аборигены из Аватара*
- элемент массива color с индексом 3, умноженный на коэффициент 138. *оно полупрозрачное и разговаривает*

Из него нужно вычесть следующее:

- weight, умноженный на коэффициент 2; *никто не любит Джаббу Хатта*
- speed, умноженный на коэффициент 5. *слишком быстро, я ничего не понял*

Если условие разумности не выполнено (sentient == false), то вау-фактор получается сложением следующих значений:

- количество символов в name, умноженное на коэффициент 23;
- weight, умноженный на speed и коэффициент 52;
- length, умноженный на коэффициент 27;
- tentacles, умноженный на коэффициент 5;
- элемент массива color с индексом 0, умноженный на коэффициент 75; *красный значит крутой*
- элемент массива color с индексом 1, умноженный на коэффициент 2.

Из него нужно вычесть следующее:

- элемент массива color с индексом 3, умноженный на коэффициент 27. *и зачем мы пришли смотреть на невидимых медуз?*

Вне зависимости от разумности, вау-фактор для хищника (carnivore == true) умножается на пять, в противном случае - делится на три.

Аргумент:

Две структуры AlienLifeForm.

Возвращаемое значение:

Одно целое число:

- положительное, если первая структура больше второй;
- ноль, если они равны;
- отрицательное, если вторая структура больше первой.

Инфраструктура (код, который можно использовать при отладке):

С экрана считывается массив структур AlienLifeForm, сортируется с помощью вашей функции сортировки, а потом на экран выводятся поля name отсортированного массива.

```
#include <iostream>
using std::cout;
using std::endl;
using std::cin;
```

```
struct AlienLifeForm
{
    char name[50];
    unsigned int id;
    unsigned int weight;
    unsigned int length;
    unsigned int speed;
    unsigned int tentacles;
    unsigned int color[4];
    bool sentient;
    bool carnivore;
};

int compare (AlienLifeForm a, AlienLifeForm b);
// ваша реализация функции
int main()
{
    int n;
    cin >> n;
    AlienLifeForm zoo[100];
    for (int i = 0; i < n; i++)
        cin >> zoo[i].name >> zoo[i].id >> zoo[i].weight
            >> zoo[i].length >> zoo[i].speed >> zoo[i].tentacles
            >> zoo[i].color[0] >> zoo[i].color[1] >> zoo[i].color[2] >> zoo[i].color[3]
            >> zoo[i].sentient >> zoo[i].carnivore;

    for (int i = 0; i < n; i++)
        for (int j = 0; j < n - i - 1; j++)
            if (compare(zoo[j], zoo[j + 1]) > 0)
            {
                AlienLifeForm tmp = zoo[j];
                zoo[j] = zoo[j + 1];
                zoo[j + 1] = tmp;
            }

    for (int i = 0; i < n; i++)
        cout << zoo[i].name << " ";
    cout << endl;
    return 0;
}
```

Внимание! Сдавать нужно только реализацию функции compare и ваши вспомогательные функции. Функцию main сдавать не нужно. Выводить что-либо на экран не нужно. Спасибо за внимание.

Например:

| Ввод | Результат |
|---|---------------------------------|
| 5 Alien 23421 32 5 26 1 0 0 0 255 0 1 Predator 34645 20 2 13 50 50 100 0 0 1 1 ALF 25344 9 1 3 0 50 50 0 255 1 1 Solaris 32534 3400543 7963135 0 0 255 0 0 128 1 0 Moth 2342 0 0 200 0 255 255 255 255 0 0 | Moth Predator ALF Alien Solaris |

Ответ: (штрафной режим: 0, 0, 0, 10, 20, ...%)

```
1 long long int wow_factor(AlienLifeForm a){ // long long int на слу
2 long long int a_wow = 0;
3 int l = 0;
4 for (int i = 0; i < 50; i++){
5     if (a.name[i] != '\0') l += 1;
6     else break;
7 }
8 //cout << l << a.name << endl;
9 if (a.sentient == true){ // если разумный
10     a_wow = a_wow + l * 126 + a.length * 5 + a.tentacles * 97
11     a_wow = a_wow - (a.weight * 2 + a.speed * 5);
12 }
13 else{ // если неразумный
14     a_wow = a_wow + (l * 23 + a.weight * a.speed * 52 + a.leng
15     a_wow = a_wow - (a.color[3] * 27);
16 }
17 if (a.carnivore == true) // если хищник
18     a_wow *= 5;
19 else
20     a_wow /= 3;
21 return a_wow;
22 }
```

| | Ввод | Ожидается | Получил | |
|---|---|---------------------------------|---------------------------------|---|
| ✓ | 5 Alien 23421 32 5 26 1 0 0 0 255 0 1 Predator 34645 20 2 13 50 50 100 0 0 1 1 ALF 25344 9 1 3 0 50 50 0 255 1 1 Solaris 32534 3400543 7963135 0 0 255 0 0 128 1 0 Moth 2342 0 0 200 0 255 255 255 255 0 0 | Moth Predator ALF Alien Solaris | Moth Predator ALF Alien Solaris | ✓ |

| | | | |
|--|-----------------|------------------|----------------|
| | 255 255 255 0 0 | | |
| | Ввод | Ожидается | Получил |
| | | | |

| | | | | |
|---|--|---|---|---|
| ✓ | 6 Scaramouch 13827 10 7 72 0 255 0 0 255 1 0 Fandango 6385 250 2 16 2 255 255 0 255 0 0 Galileo 1737 12 5 2 0 15 27 184 255 1 0 Figaro 6281 8 3 24 5 82 16 93 100 0 0 Bismillah 38401 47 30 7 0 255 255 255 0 0 1 Beelzebub 42 1 1 5 0 0 0 0 255 1 1 | Figaro Scaramouch Galileo Fandango Beelzebub Bismillah | Figaro Scaramouch Galileo Fandango Beelzebub Bismillah | ✓ |
| ✓ | 4 Lashunta 33641 7 2 20 4 0 0 255 0 1 0 Kasatha 33741 8 3 25 6 128 128 128 0 1 0 Shirren 33841 6 2 30 50 128 0 255 0 1 1 Vesk 33941 13 3 15 4 0 255 0 0 0 0 | Kasatha Lashunta Vesk Shirren | Kasatha Lashunta Vesk Shirren | ✓ |
| ✓ | 9 Quelaag 23717 140 5 10 8 255 0 0 120 1 0 Sif 1231 167 8 37 0 85 85 85 255 0 1 Vordt 26482 215 3 25 0 124 124 124 255 0 0 Yhorm 76234 532 10 8 0 24 66 66 255 1 0 Oceiros 7862 100 6 8 3 105 105 105 255 1 1 Aldrich 4573 304 10 3 10 12 43 23 178 1 1 Nashandra 236 75 3 5 2 0 0 0 255 1 0 Velstadt 348975 120 2 7 0 31 52 140 200 1 0 Vendrick 9834 230 4 5 0 32 64 124 255 0 0 | Quelaag Velstadt Nashandra Yhorm Vendrick Vordt Aldrich Oceiros Sif | Quelaag Velstadt Nashandra Yhorm Vendrick Vordt Aldrich Oceiros Sif | ✓ |

Все тесты пройдены! ✓

Решение автора вопроса:

```
int wow(AlienLifeForm a)
{
    int n = 0;
    while (a.name[n] != '\0')
        n++;
    int res = 0;
    if (a.sentient)
        res += 126 * n - 2 * a.weight - 5 * a.speed + 5 * a.length + 97 * a.tentacles + 4 *
a.color[0] + 4 * a.color[1] + 30 * a.color[2] + 138 * a.color[3];
    else
        res += 23 * n + 52 * a.weight * a.speed + 27 * a.length + 5 * a.tentacles + 75 * a.color[0]
+ 2 * a.color[1] - 27 * a.color[3];
    if (a.carnivore)
        res *= 5;
    else
        res /= 3;
    return res;
}

int compare (AlienLifeForm a, AlienLifeForm b)
{
    return wow(a) - wow(b);
}
```

Верно
Баллы за эту попытку: 1,00/1,00.

Вопрос **6**
 Верно
 Баллов: 1,00 из 1,00

Репликанты

Рой и Прис играли в крестики-нолики, но их отвлекли. Сделайте последний ход за Роя.

Дана недоигранная партия в крестики-нолики. Поле обычное, 3 на 3, задается с экрана символами.

Нужно найти ход, который выигрывает игру за "крестики", и вывести окончательный вид игрового поля.

Если такого хода нет, то вывести -1.

Гарантируется, что поле является корректным, а правильный ход - единственным.

Входные данные:

Начальный вид игрового поля. Набор из 9 символов: 'x' - крестик, 'o' - нолик, ' ' (пробел) - пустая клетка. Каждая строка заканчивается символом конца строки '\n'.

Выходные данные:

Окончательный вид игрового поля в том же формате, что и ввод.

Например:

| Ввод | Результат |
|-----------------|------------------|
| xx oo xхо | xxx oo xхо |

Ответ: (штрафной режим: 0, 0, 0, 10, 20, ...%)

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6      int a[3][3] = {0}, s = 0; // матрица 3 на 3 (фактически наше и
7      int x; char k; // i это строка j это столбец
8      for (int i = 0; i < 3; i++){ // считываем ввод
9          for (int j = 0; j < 3; j++){
10             k = cin.get();
11             if (k == ' ') // но заменяем его числами (с ними проще
12                 x = 0;
13             else if (k == 'x')
14                 x = 1;
15             else if (k == 'o')
16                 x = 1000;
17             a[i][j] = x;
18         }
19         k = cin.get();
20     }
21     bool endeth = false; // переменная "конца" поиска. флаг. прини
22
    
```

| | Ввод | Ожидается | Получил | |
|---|-------------------|-------------------|-------------------|---|
| ✓ | xx oo xхо | xxx oo xхо | xxx oo xхо | ✓ |
| ✓ | x охо | x охо x | x охо x | ✓ |
| ✓ | xхо хох oo | xхо хох хоо | xхо хох хоо | ✓ |
| ✓ | oo xx | oo xxx | oo xxx | ✓ |
| ✓ | хох o o хох | хох охо хох | хох охо хох | ✓ |

Все тесты пройдены! ✓

Решение автора вопроса:

```

#include <iostream>
using namespace std;

int main()
{
    char a[3][3];
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
            a[i][j] = cin.get();
        cin.get();
    }

    for (int i = 0; i < 3; i++)
    {
        // checking the i-th line
        int crosses = 0, zeroes = 0, empty = 0;
        for (int j = 0; j < 3; j++)
        {
            if (a[i][j] == 'x') crosses++;
            if (a[i][j] == 'o') zeroes++;
            if (a[i][j] == ' ') empty = j;
        }
        if (crosses == 2 && zeroes == 0)
        {
            a[i][empty] = 'x';
            // the way of printing the result will be a nice material for antiplagiat
            for (int i = 0; i < 3; i++)
            {
                for (int j = 0; j < 3; j++)
                    cout << a[i][j];
                cout << endl;
            }
            return 0;
        }

        // checking the i-th column
        crosses = 0, zeroes = 0, empty = 0;
        for (int j = 0; j < 3; j++)
        {
            if (a[j][i] == 'x') crosses++;
            if (a[j][i] == 'o') zeroes++;
            if (a[j][i] == ' ') empty = j;
        }
        if (crosses == 2 && zeroes == 0)
        {
            a[empty][i] = 'x';
            for (int i = 0; i < 3; i++)
            {
                for (int j = 0; j < 3; j++)
                    cout << a[i][j];
                cout << endl;
            }
            return 0;
        }
    }

    // checking diagonals
    int crosses = 0, zeroes = 0, empty = 0;
    for (int i = 0; i < 3; i++)
    {
        if (a[i][i] == 'x') crosses++;
        if (a[i][i] == 'o') zeroes++;
        if (a[i][i] == ' ') empty = i;
    }
    if (crosses == 2 && zeroes == 0)
    {
        a[empty][empty] = 'x';
        for (int i = 0; i < 3; i++)
        {
            for (int j = 0; j < 3; j++)
                cout << a[i][j];
            cout << endl;
        }
        return 0;
    }

    // we get here if there was no winning turn
    cout << -1 << endl;
    return 0;
}

```

Верно

Баллы за эту попытку: 1,00/1,00.

Вопрос **7**

Нет ответа

Баллов: 0,00 из 1,00

Хаджит и рыбы

Старый хаджит ловит рыбу на холодном и уютном побережье Моря Призраков. Хитрый рыбак опытен, он не бежит сломя голову, нет. Для начала он залезает на скалу повыше, разводит маленький бездымный костерок для завтрака и смотрит, что происходит в бухточках и фьордах. Где плавает косяк рыбы побольше, а где сидят разбойники, и сколько ему придется заплатить, чтобы они оставили его в покое. Хаджит знает свои силы - за один переход он сможет перейти не больше трех холмов между бухтами - и планирует путь соответственно. Это будет еще один непростой день. Но даже если сегодня ему не повезло, у него будут новые дни и будут вкусные рыбы. Хаджит потягивается, похрустывая каждым позвонком, и щурится на восходящее солнце. Пора в дорогу.

Игровое поле состоит из одномерного массива клеток. В каждой клетке написано количество получаемых очков (может быть отрицательным). Игрок может делать шаг вперед на одну, две или три клетки. Соответственно, первым ходом он идет на первую, вторую или третью клетку. Последним ходом он выходит с игрового поля. Если первый ход совпадает с последним, то игрок имеет право не заходить на игровое поле и получить 0 очков. Плохое побережье, мутсера. Хаджит не пойдет туда, о нет.

Нужно максимизировать количество очков.

Входные данные:

Целое число n - количество клеток (1 <= n <= 100).

Набор из n целых чисел, каждое число - это количество очков, получаемых в соответствующей клетке (от -1000 до 1000).

Выходные данные:

Одно целое число - максимальное количество очков, которое можно набрать на этом игровом поле.

Например:

| Ввод | Результат |
|-------------------------------|-----------|
| 10 1 2 3 -1 -2 -3 -2 3 3 2 | 12 |
| 5 -5 -5 -5 1 1 | -3 |

Ответ: (штрафной режим: 0, 0, 0, 10, 20, ...%)

1 ||

Решение автора вопроса:


```
#include <iostream>
using namespace std;

int max(int a, int b, int c)
{
    int max = a > b ? a : b;
    max = c > max ? c : max;
    return max;
}

int main()
{
    int n, a[100], b[100];
    cin >> n;
    for (int i = 0; i < n; i++)
        cin >> a[i];
    if (n == 1)
    {
        cout << ( a[0] > 0 ? a[0] : 0 ) << endl;
        return 0;
    }
    else if (n == 2)
    {
        cout << max(a[0] + a[1], a[1], 0) << endl;
        return 0;
    }

    b[0] = a[0];
    b[1] = a[0] + a[1] > a[1] ? a[0] + a[1] : a[1];
    b[2] = max(a[0] + a[2], b[1] + a[2], a[2]);

    if (n == 3)
    {
        cout << b[2] << endl;
        return 0;
    }

    for (int i = 3; i < n; i++)
        b[i] = max(b[i - 3], b[i - 2], b[i - 1]) + a[i];

    cout << max(b[n - 3], b[n - 2], b[n - 1]) << endl;
    return 0;
}
```

Вопрос **8**

Нет ответа

Баллов: 0,00 из 1,00

ascii-art

Вы летите в гиперканале и готовитесь выйти обратно в обычный космос. Станция инопланетян, к которой летит ваш корабль с дипломатической миссией, очень высокого мнения о способностях навигатора и пилота - они считают, что вы сможете попасть из гиперканала прямо в док (сложной, зараза, формы) и ничего им там не разнести. Они уже рассчитали и передали вам координаты вершин полигона, которые представляют собой проекцию дока на сечение гиперканала. Пилот, наверное, справится. Вы ему только визуализируйте, куда рулить.

Замкнутая ломаная на плоскости задаётся упорядоченным набором из N точек.

Каждая пара соседних точек образует ребро ненулевой длины.

Например, набор точек (a, b, c) образует треугольник с рёбрами (a, b), (b, c), (c, a).

Ни одно ребро не пересекает никакое другое.

Сами точки задаются парами целочисленных координат (x, y), по модулю не превышающих 2¹⁰.

Описанная ломаная высекает из плоскости полигон.

Нужно составить из этого полигона простой ascii-art.

А именно, для каждой точки с целочисленными координатами (x, y) такими, что x_{min} ≤ x ≤ x_{max}, y_{min} ≤ y ≤ y_{max}, нужно вывести точку, если эта точка лежит вне полигона -- иначе, вывести ноль.

Считать, что точки, принадлежащие ломаной, не лежат вне полигона.

Входные данные:

Четыре целых числа - координаты x_{min}, y_{min}, x_{max}, y_{max}.

Одно целое число N - количество точек в ломаной (3 ≤ N ≤ 100).

N пар целых чисел - координаты точек ломаной.

Выходные данные:

Набор символов (см. примеры). Для каждой точки в прямоугольнике вывести 0, если она внутри полигона и точку ('.'), если снаружи.

Например:

| Ввод | Результат |
|-----------|-----------------------|
| 0 0 15 10 | |
| 3 |0..... |
| 1 1 |000..... |
| 9 9 |00000..... |
| 14 4 |0000000..... |
| |000000000..... |
| |00000000000..... |
| |0000000..... |
| |0000..... |
| |0..... |
| | |

Ответ: (штрафной режим: 0, 0, 0, 10, 20, ...%)

1 ||

Решение автора вопроса:

```

struct vec2
{
    int x, y;
};

#include <iostream>
#include <vector>
int main()
{
    vec2 pmin, pmax;
    std::cin >> pmin.x >> pmin.y;
    std::cin >> pmax.x >> pmax.y;

    unsigned int n;
    std::cin >> n;

    std::vector<vec2> point;
    point.reserve(n);

    vec2 p;
    for(unsigned int i = 0u; i < n; ++i)
    {
        std::cin >> p.x >> p.y;
        point.push_back(p);
    }

    for(int py = pmax.y; py >= pmin.y; --py)
    {
        for(int px = pmin.x; px <= pmax.x; ++px)
        {
            vec2 p = {px, py};

            // https://en.wikipedia.org/wiki/Point_in_polygon
            // cast a ray from (p.x, p.y) in positive-x direction, count winding number
            // counting intersection number will fail when ray crosses the polygon vertex

            int wn = 0;
            int wnv = 0; // separate winding number for vertices -- count them twice, divide by two at the
end
            for(unsigned int j = 0u; j < n; ++j)
            {
                vec2 const p0 = point[ j ];
                vec2 const p1 = point[(j + 1) % n];
                vec2 const dp1 = {p1.x - p0.x, p1.y - p0.y};
                vec2 const dp = {p.x - p0.x, p.y - p0.y};

                if(dp1.y != 0)
                {
                    double const t = double(dp.y) / double(dp1.y);
                    double const x = p0.x + t * dp1.x;
                    if(t < 0. || t > 1. || x < p.x)
                        continue;
                    if(x == p.x)
                    {
                        wn = -1;
                        wnv = 0;
                        break;
                    }
                    int const dwn = dp1.y > 0 ? 1 : -1;
                    if(t == 0. || t == 1.)
                        wnv += dwn;
                    else
                        wn += dwn;
                }
                else if(p.y == p0.y)
                {
                    double const t = double(dp.x) / double(dp1.x);
                    if(t >= 0. && t <= 1.)
                    {
                        wn = -1;
                        wnv = 0;
                        break;
                    }
                }
            }
            std::cout << (wn + wnv / 2 != 0 ? '0' : '.');
        }
        std::cout << std::endl;
    }
}

```