

# Лабораторная работа №1 (executor)

## Содержание

1. [Содержание](#)
2. [Задание](#)
  1. [Советы](#)
3. [Требования к корректности решения](#)
  1. [Базовые требования](#)
  2. [Дополнительные требования](#)
  3. [Цитатник](#)
  4. [Бродилка](#)
  5. [Стенки](#)
  6. [Аккуратное чтение](#)
4. [Инструкция по сдаче](#)
5. [Система оценки](#)
6. [Сроки сдачи](#)

## Задание

Это задание состоит из четырёх частей, вы можете выполнить только префикс и [получить частичные баллы](#).

Вы пишете интерпретатор команд для некоторого исполнителя.

Решением является один файл `executor.cpp`, каждая следующая часть расширяет предыдущие.

1. **Цитатник.**

По командам `sherlock-holmes` и `cat-j-eoffry` выведите на экран отрывок из Шерлока Холмса и Jubilate Agno, соответственно.

В точности сохраните все пробельные символы, включая переводы строк.

По команде `quit` успешно завершите интерпретатор.

У команд могут быть псевдонимы.

Гарантируется, что входные данные корректны.
2. **Бродилка.**

Проэмулируйте бродилку по бесконечной квадратной сетке: исходно вы находитесь в клетке `(0, 0)` и смотрите вверх.

Команды позволяют ходить вперёд, назад, поворачиваться на 90 градусов влево и направо, а также показывать исследованную карту при помощи [ASCII графики](#).

Также должна поддерживаться запись произвольной последовательности команд, чтобы её потом можно было просто выполнить.

### 3. Стенки.

Добавьте две новые команды: добавление и удаление стенки перед игроком. Если игрок пытается пройти сквозь стену, выводится сообщение `Bump!`.

### 4. Аккуратное чтение.

Входные данные теперь — произвольная последовательность байт.

Каждая команда должна располагаться на отдельной строчке, могут быть дописаны пробельные символы в начале или в конце.

Некорректные команды должны приводить к сообщению `Invalid command: '<команда>'`.

Точный формат смотрите в тестах, а семантику команд — в требованиях ниже.

## Советы

- \* Выполняйте части сверху вниз. Удалите из `tests.txt` тесты, которые вы пока что не собираетесь проходить.
- \* Никогда не используйте `while (cin)` или `while (cin.eof())` для проверки конца файла. Сначала попробуйте прочитать, а потом проверяйте успешность последнего чтения, [как тут](#).
- \* Используйте `std::getline` для построчного чтения.
- \* Помните, что `char` может быть знаковым типом, `std::isspace` и похожие функции работают с `unsigned char`, а EOF на некоторых системах равен `-1`.
- \* Для крупных типов (вроде `std::string`) используйте передачу по константной ссылке, где разумно.
- \* Вместо десятка разрозненных констант используйте `enum class` ([scoped enumeration](#)).
- \* Добавьте свои собственные тесты в `tests.txt` и запускайте их на сервере.
- \* Отдельные функции удобно тестировать в начале `int main()` при помощи `assert`.
  - Удобно выделить функцию `parse_line`, которая превращает строку со входа во что-то более простое для обработки: `enum` или `std::optional<>`.
- \* Заведите константные массивы, в которых будет храниться ожидаемый сдвиг для каждого из четырёх направлений, например:

```
struct Direction {
    int dx, dy;
};
const Direction DIRECTIONS[] = {{1, 0}, {0, 1}, {-1, 0}, {0, -1}};
int x = 10, y = 10, direction = 2;
x += DIRECTIONS[direction].dx;
y += DIRECTIONS[direction].dy;
```

- \* Не пытайтесь завести двумерный массив для всего поля. Придумайте тест, на котором после `n` операций потребуется хранить массив размера  $O(n^2)$ .
- \* Вам наверняка пригодятся `std::set`, `std::map` и `std::pair`.

- \* `clang-tidy` может ругаться на глобальные переменные. В общем случае их плохо использовать, но в этом задании разумно, так что можно заглушить соответствующие предупреждения.
- \* Если вы скачаете себе ZIP-архив с репозиториум на Windows, то в файлах с ожидаемыми ответами будут использоваться линуксовые переводы строк — один символ `\n`. А под Windows ожидается виндовый перевод строк (два символа: `\r\n`), это автоматически обрабатывается стандартной библиотекой: вы выводите `\n`, а она преобразует это в `\r\n`. Аналогично с чтением. Так как редакторы обычно подстраиваются под стиль переводов строк, ответом и ваш вывод будет выглядеть одинаково, но команда `diff` и локальные тесты будут считать, что они отличаются во всех строчках. Это чинится преобразованием переводов строк во всех файлах при помощи команды `unix2dos`, которую можно поставить в `msys2` командой `pacman -S dos2unix`.

## Требования к корректности решения

### Базовые требования

Действуют [стандартные требования](#).

### Дополнительные требования

$n$  операций должны работать не медленнее чем  $O(n \log n + A)$  амортизированно, где  $A$  — размер вывода программы.

### Цитатник

Чтение команд идёт со стандартного потока ввода (`stdin`), вывод всех результатов — в стандартный поток вывода (`stdout`).

Каждая команда располагается на отдельной строке.

Символов, кроме команд и переводов строки, нет.

Поддерживаемые команды:

Псевдонимы	Поведение
<code>quit, exit</code>	Немедленное успешное завершение интерпретатора
<code>sherlock, sherlock-holmes</code>	Вывод на экран цитаты из Шерлока Холмса
<code>cat, cat-jeoffry</code>	Вывод на экран цитаты из Jubilate Agno

Цитата из Шерлоха Холмса:

I had had so many reasons to believe in my friend's subtle powers of reasoning and extraordinary energy in action that I felt that he must have some solid grounds for the assured and easy demeanour with which he treated the singular mystery which he had been called upon to fathom. Once only had I known him to fail, in the case of the King of Bohemia and of the Irene Adler photograph; but when I looked back to

the weird business of the Sign of Four, and the extraordinary circumstances connected with the Study in Scarlet, I felt that it would be a strange tangle indeed which he could not unravel.

Цитата из Jubilate Agno:

For he is of the tribe of Tiger.  
For the Cherub Cat is a term of the Angel Tiger.  
For he has the subtlety and hissing of a serpent, which in goodness he suppresses.  
For he will not do destruction, if he is well-fed, neither will he spit without provocation.  
For he purrs in thankfulness, when God tells him he's a good Cat.  
For he is an instrument for the children to learn benevolence upon.  
For every house is incomplete without him and a blessing is lacking in the spirit.  
For the Lord commanded Moses concerning the cats at the departure of the Children of Israel from Egypt.  
For every family had one cat at least in the bag.  
For the English Cats are the best in Europe.

Первую цитату следует написать в программе при помощи одного или нескольких обычных строковых литералов: "HeI Io".

Вторую цитату следует написать при помощи сырых (raw) [строковых литералов](#): R"(HeI Io)".

## Бродилка

Поддерживаемые команды:

Псевдонимы	Поведение
<code>forward, fwd</code>	Пройти на одну клетку вперёд
<code>backward, bwd</code>	Пройти на одну клетку назад
<code>turn-right, right</code>	Повернуться на 90 градусов направо, по часовой стрелке
<code>turn-left, left</code>	Повернуться на 90 градусов налево, против часовой стрелки
<code>show-map</code>	Выводит на экран текущую карту
<code>start-macro</code>	Начинает запись макроса (смотри ниже)
<code>run-macro</code>	Запускает записанный ранее макрос

## Формат вывода карты

Найдём минимальный прямоугольник с осями, параллельными осям координат, содержащий все посещённые игроком клетки.

Пусть он имеет размер  $W$  клеток в ширину и  $H$  клеток в высоту.

Тогда карта выводится как  $2*H+1$  строка, содержащая  $2*W+1$  символ каждая.

Чётные символы в чётных строках (при нумерации с единицы) соответствуют клеткам:

- \* `.` — клетка ещё не была посещена игроком;
- \* `o` — клетка была посещена игроком;

\* `^/>>/v/<` — игрок сейчас находится в этой клетке и смотрит вверх/направо/вниз/налево.

Остальные символы в этом задании должны быть пробельными, точный формат смотрите в открытых тестах.

В следующем задании там появятся стенки между клетками.

## Макрос

Интерпретатор помнит ровно один макрос — последний записанный.

Исходно макрос пустой.

При записи макроса команды продолжают читаться так же, как и раньше, но вместо выполнения они записываются в макрос

и выводят на экран сообщение `Recorded`.

В макрос можно записать только команды `forward`, `backward`, `turn-right`, `turn-left`, `show-map` и их псевдонимы.

Остальные команды во вводе не возникают.

Запись макроса заканчивается командой `end-macro` или концом файла, после чего на экран выводится сообщение

`New macro of X command(s)`, где `X` — количество команд в макросе.

В обычном режиме можно запустить макрос командой `run-macro`.

Она выполняется так же, как если бы команды из макроса ввели вручную.

## Стенки

Добавьте ещё две команды, в том числе в макросы:

Псевдонимы	Поведение
<code>put-wall</code>	Поставить стенку перед игроком, если она есть — ничего не делать
<code>remove-wall</code>	Удалить стенку перед игроком, если её нет — ничего не делать

Стенки ставятся между клетками.

Каждая стенка загораживает проход между двумя соседними клетками.

Если игрок пытается пройти сквозь стену, то у него это не должно получиться и интерпретатор выводит сообщение `Bump!`.

Стены также должны выводиться на карте.

Для вертикальных стен между клетками используйте `|`, для горизонтальных — `-` (дефис).

В углах клетки ставьте `+` тогда и только когда, когда у этого угла есть хотя бы одна смежная стена.

## Аккуратное чтение

Определим формально формат ввода:

1. Ввод разделяется на строки по стандартным конвенциям POSIX: строка (line) — это последовательность байт (возможно, пустая), завершённая ровно одним символом перевода строки (`\n`, код 10).

Также может быть одна непустая строка (incomplete line) в конце ввода: непустая последовательность байт, после которой вместо `\n` идёт конец ввода.

- Обработка ОС-специфичных переводов строки (`\r\n` под Windows, `\r` под старыми Mac) выполняется стандартной библиотекой языка C++, вам не нужно про это думать.  
В частности, во входном файле при тестировании на Windows перед любым байтом `\n` всегда будет идти байт `\r`.

2. В каждой строке может находиться одно из двух:

- Команда, возможно, с добавлением пробельных символов (whitespace characters) до или после неё.  
В таких строках интерпретатор выполняет соответствующую команду.  
Пробельные символы берутся только из таблицы ASCII.
- Любая другая последовательность байт.  
В таких строках интерпретатор выдаёт сообщение `Invalid command: '<последовательность-байт>'`.

В частности, строки могут быть пустыми или состоять только из пробельных символов. Они вызывают такое же сообщение об ошибке, как и остальные, начинающееся с `Invalid command`.

Если внутри макроса пытаются записать команду, которая макросами не поддерживается, возникает такое же сообщение `Invalid command`, даже если команда была корректны вне макроса.