

Яндекс Такси

C++ на практике

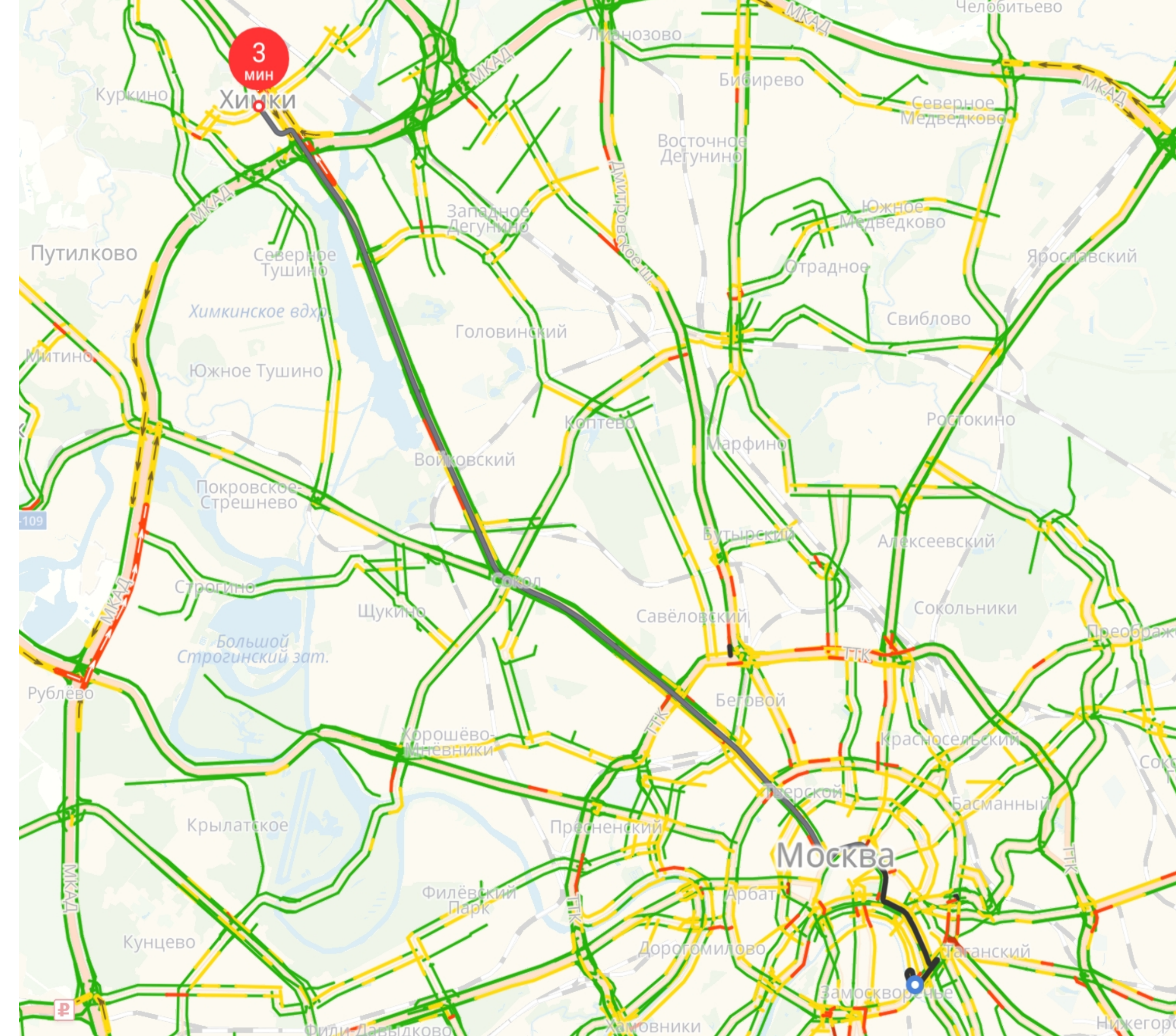
Полухин Антон

Antony Polukhin

Яндекс Такси

Содержание

- Введение
- Java не тормозит
- Чего не хватает



- Java

- C++

Подъезд

ЭКОНОМ
4Р


КОМФОРТ
8₽

 **КОМФОРТ+**
9₽



БИЗНЕС
34₽



МИНИВЭН
15Р

ДЕТСКИЙ
2Р

Комментарий, пожелания

Способ оплаты
Команда Яндекс.Такси

Введение

(или о WG21)

C++ - это

компилируемый, статически типизированный
язык программирования общего назначения.

C++ - это

компилируемый, статически типизированный
язык программирования **общего назначения**.

C++ - это

компилируемый, статически типизированный язык программирования **общего назначения**.

Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также развлекательных приложений (игр).

Устроим мини испытание

Испытание

Написать новогоднюю программу с ASCII art анимацией

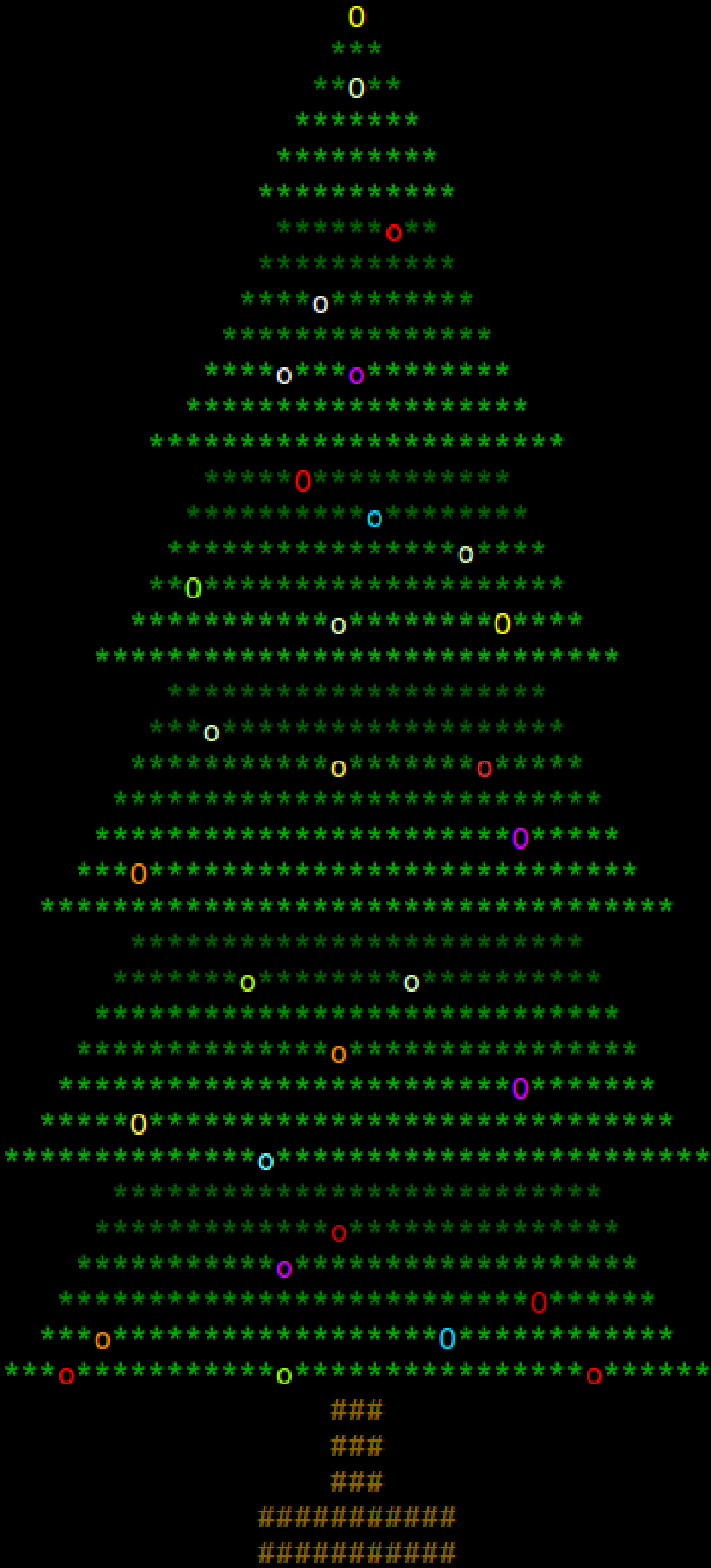
Испытание

Написать новогоднюю программу с ASCII art анимацией

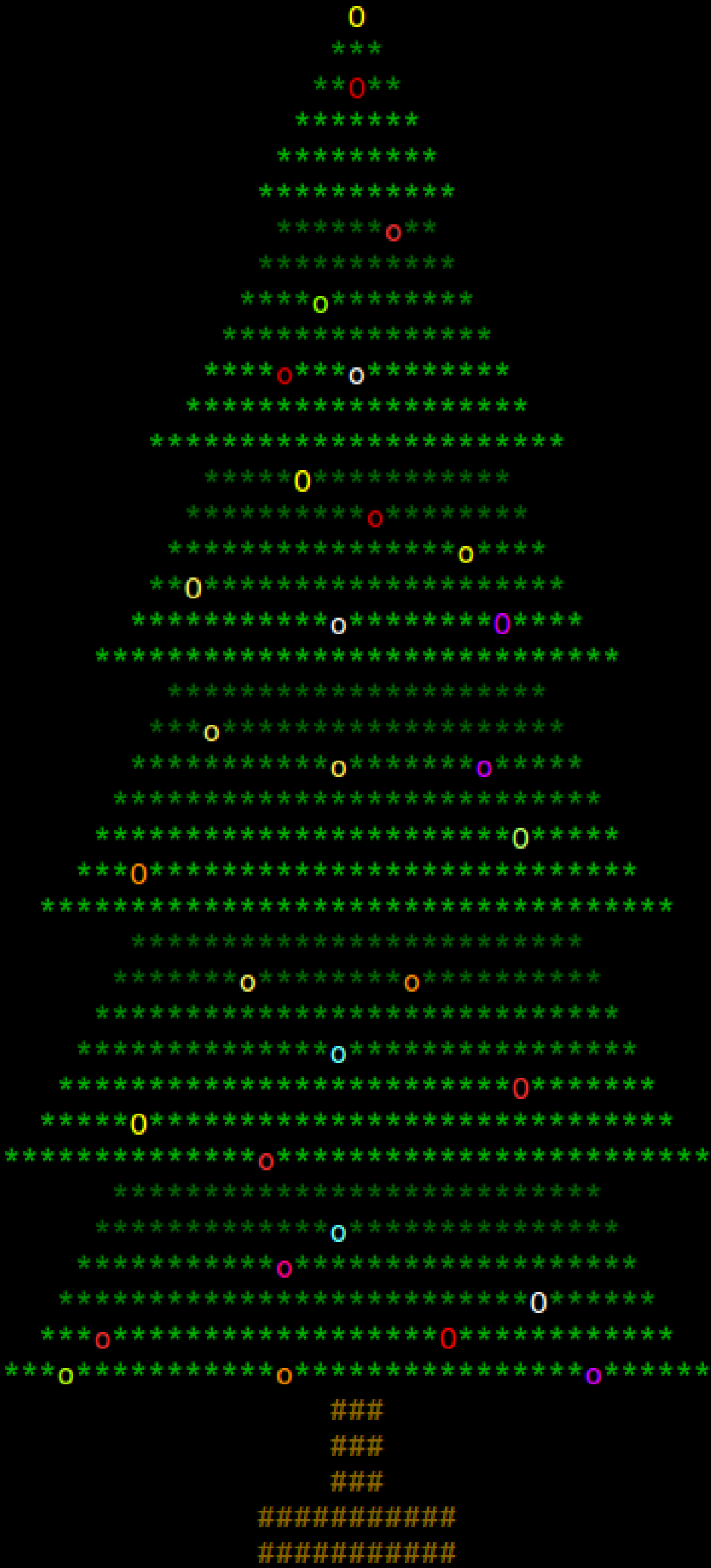
<https://github.com/apolukhin/christmas-tree>

С НОВЫМ ГОДОМ!

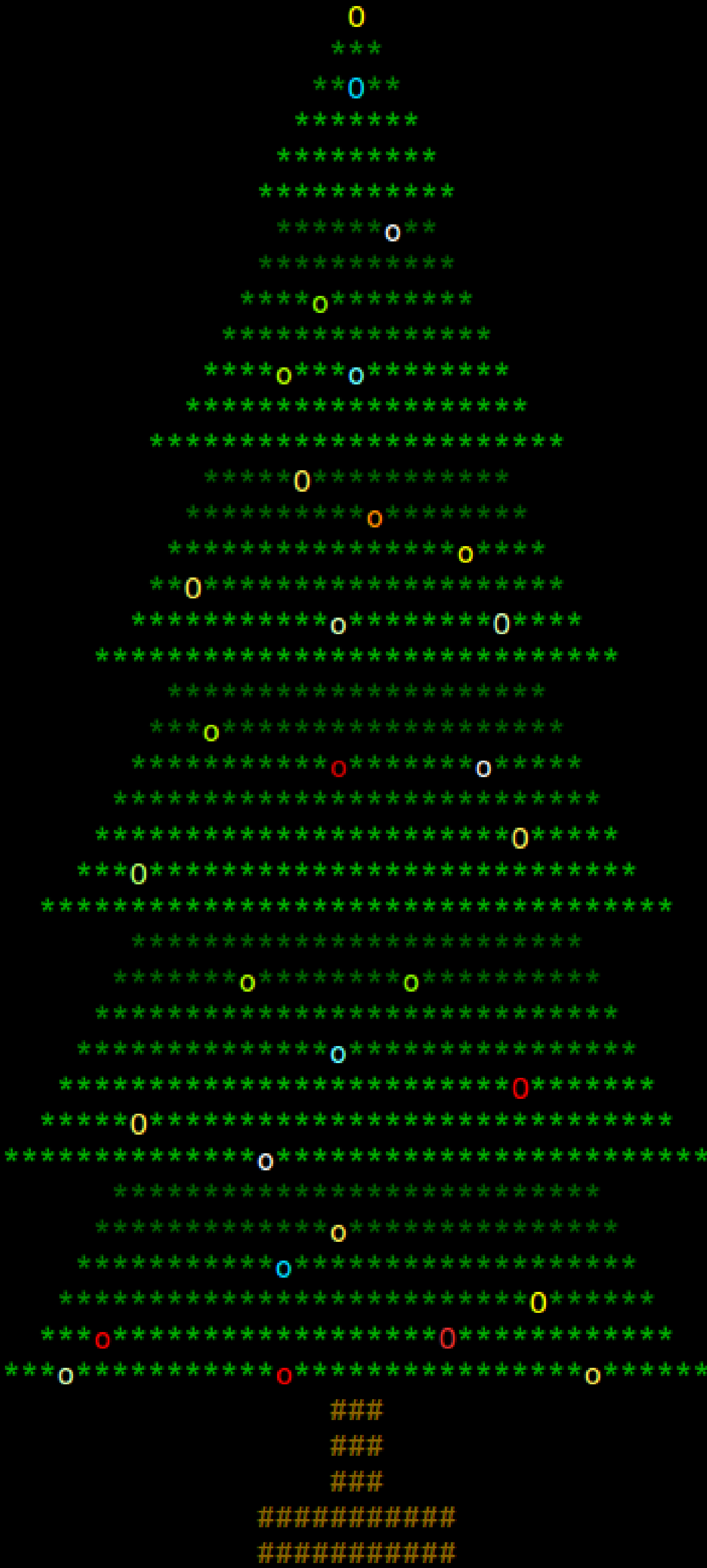
С НОВЫМ ГОДОМ!



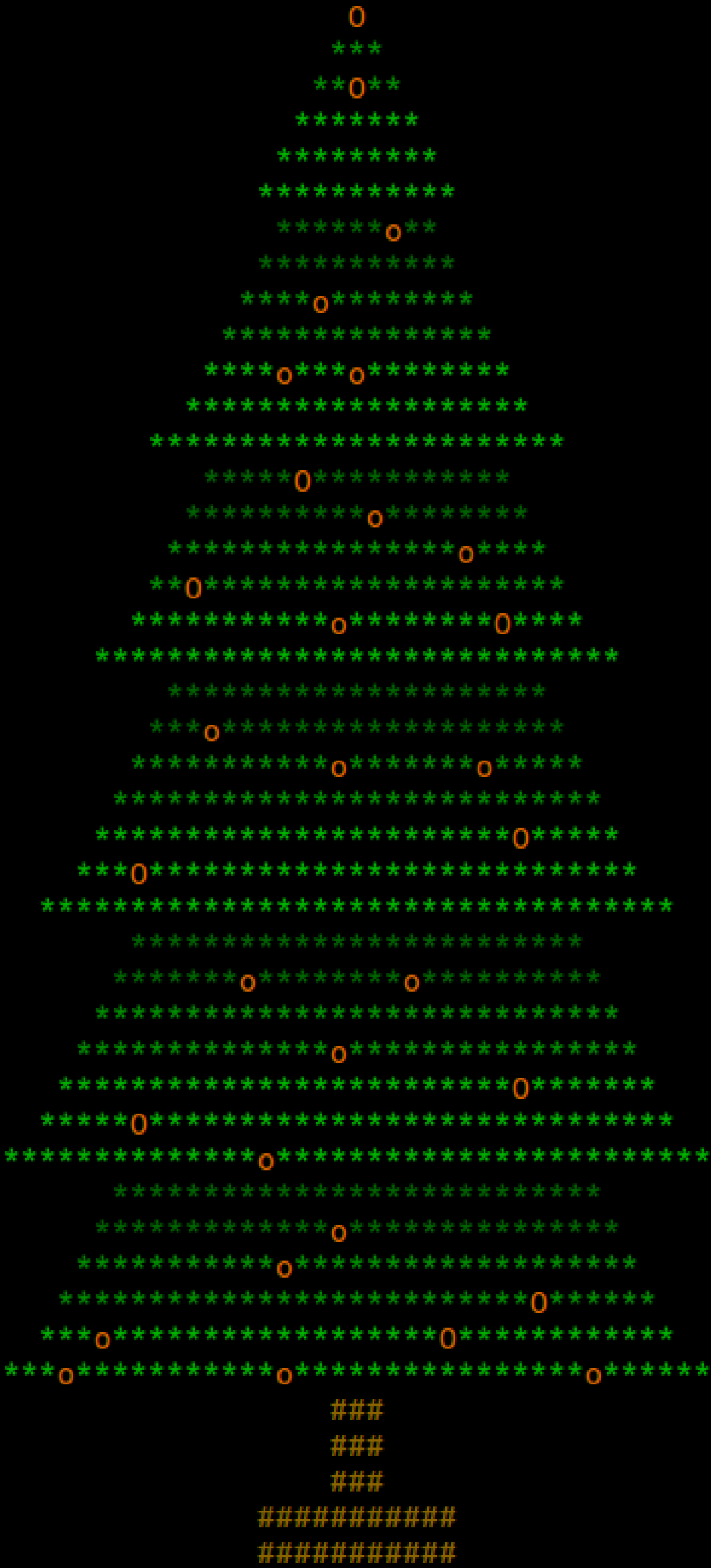
С НОВЫМ ГОДОМ!



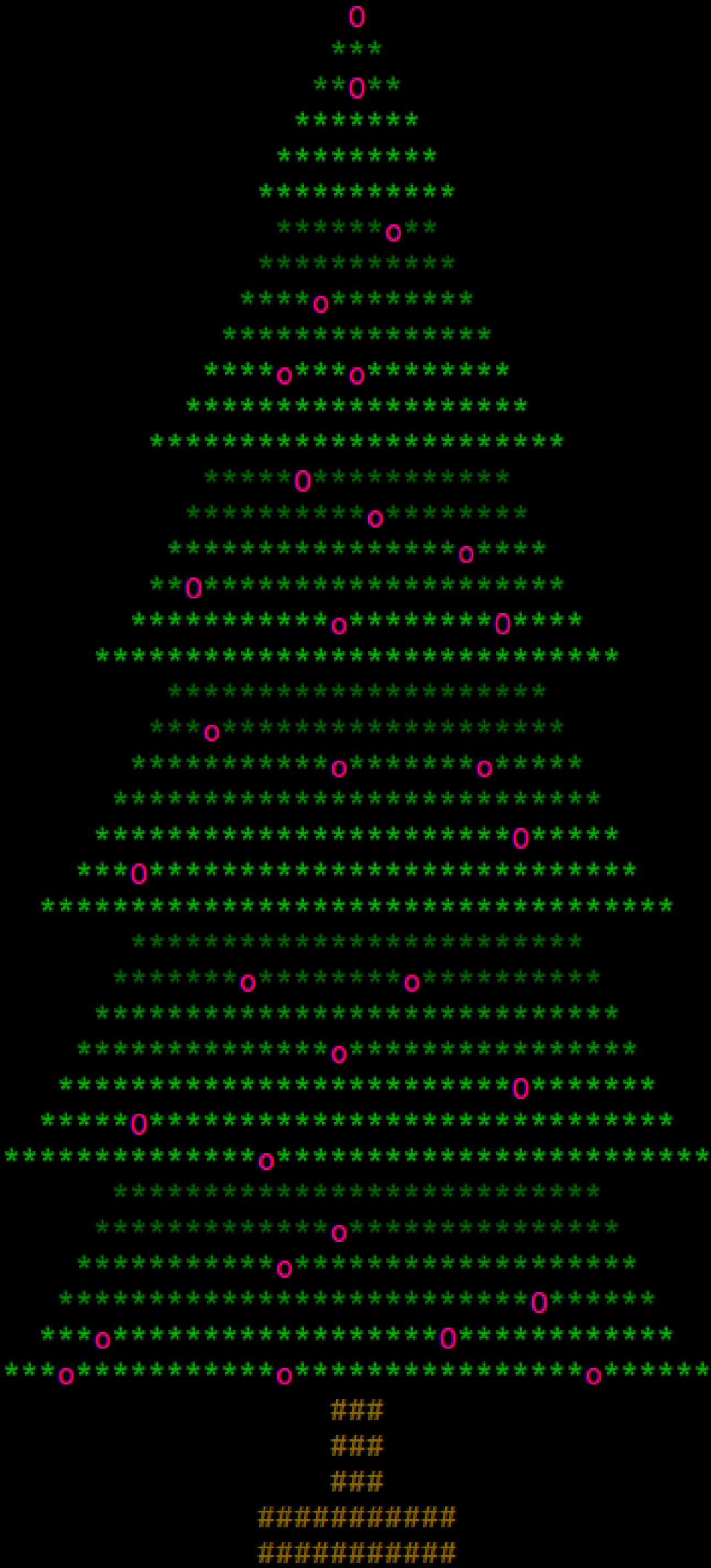
С НОВЫМ ГОДОМ!



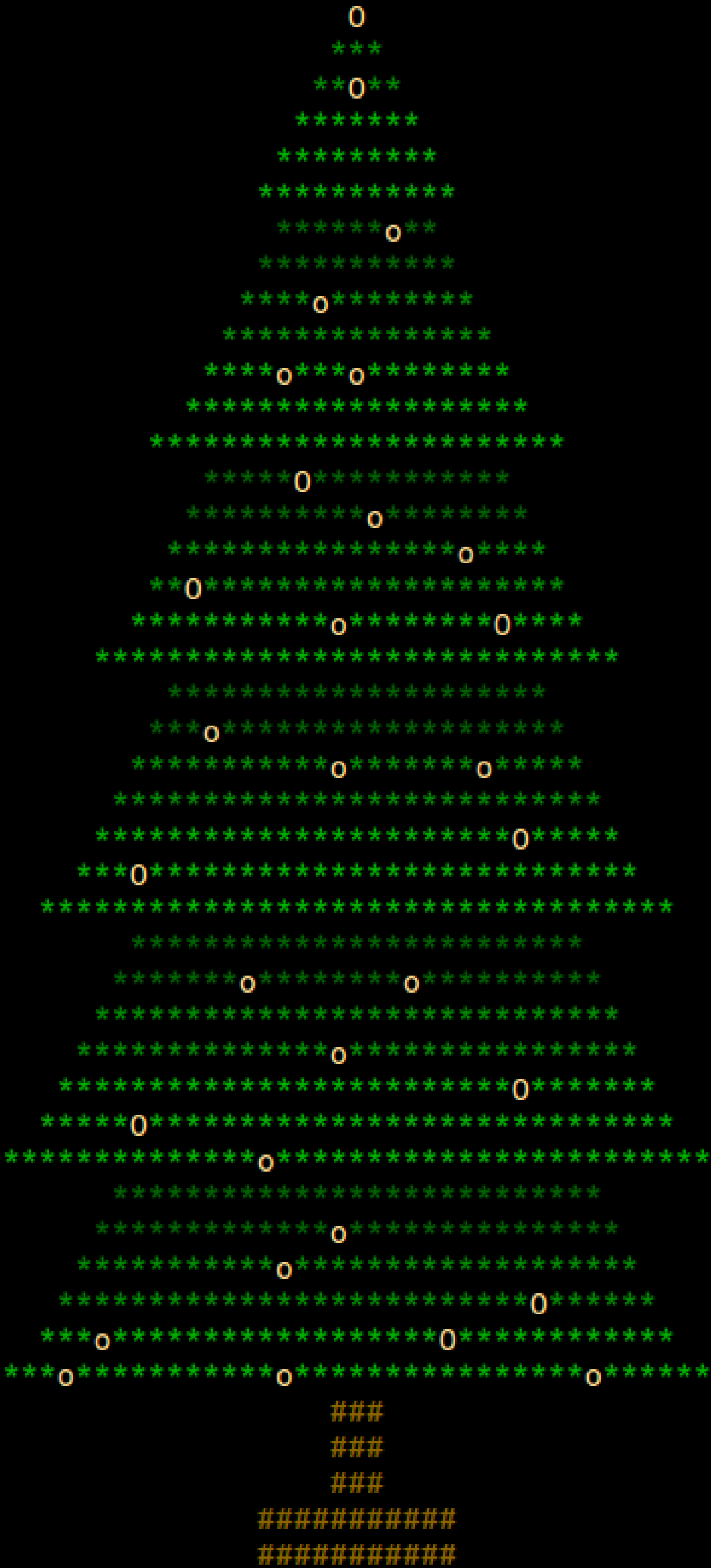
С НОВЫМ ГОДОМ!



С НОВЫМ ГОДОМ!



С НОВЫМ ГОДОМ!



Исходники ёлочки (некрасивые части)

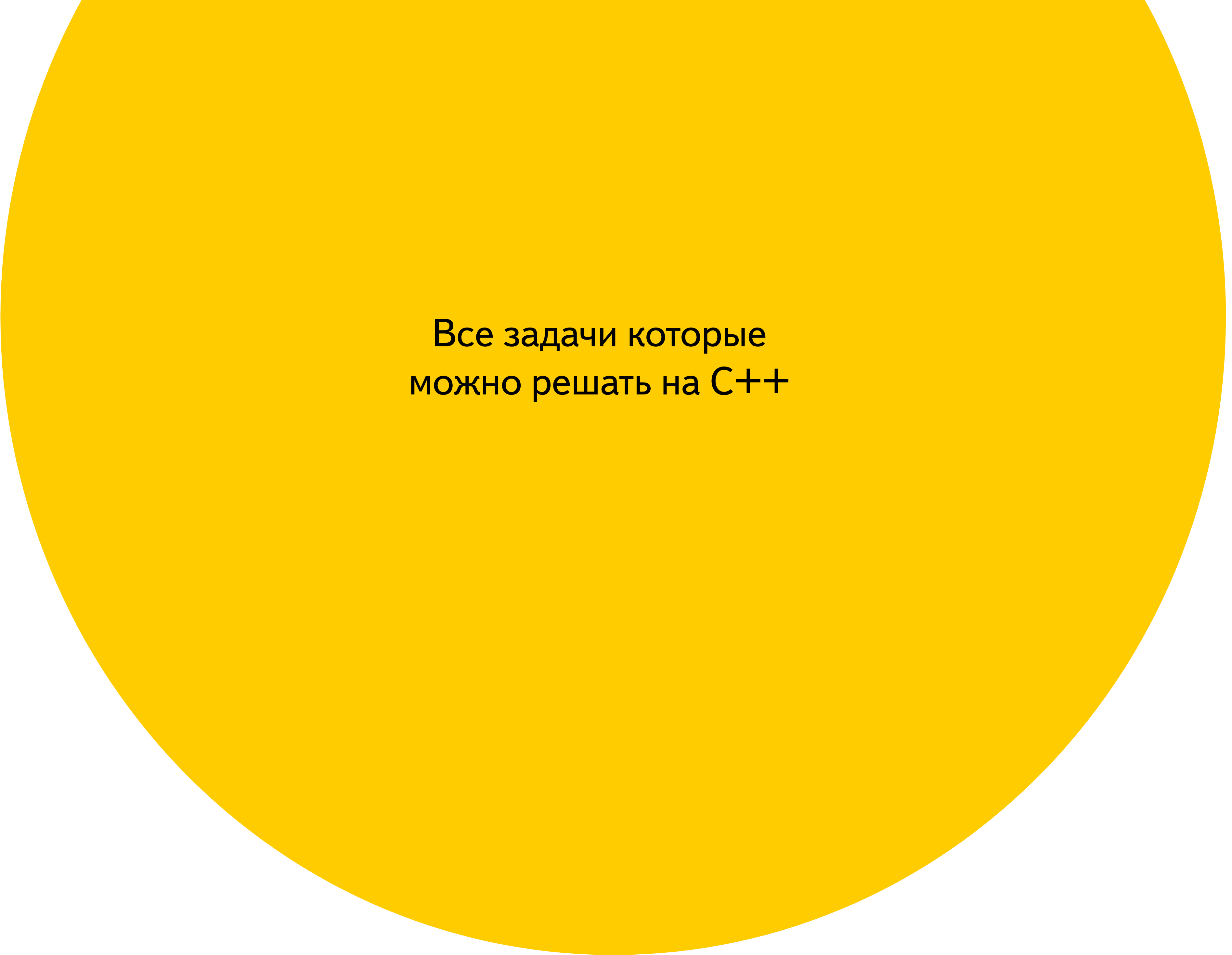
Исходники ёлочки (некрасивые части)

```
std::thread t([&lamp]() {  
    char c;  
    while (std::cin >> c) {  
        lamp.change_mode();  
    }  
});
```

Исходники ёлочки (некрасивые части)

```
std::ifstream ifs{filename.c_str()};  
std::string tree;  
std::getline(ifs, tree, '\\0');
```

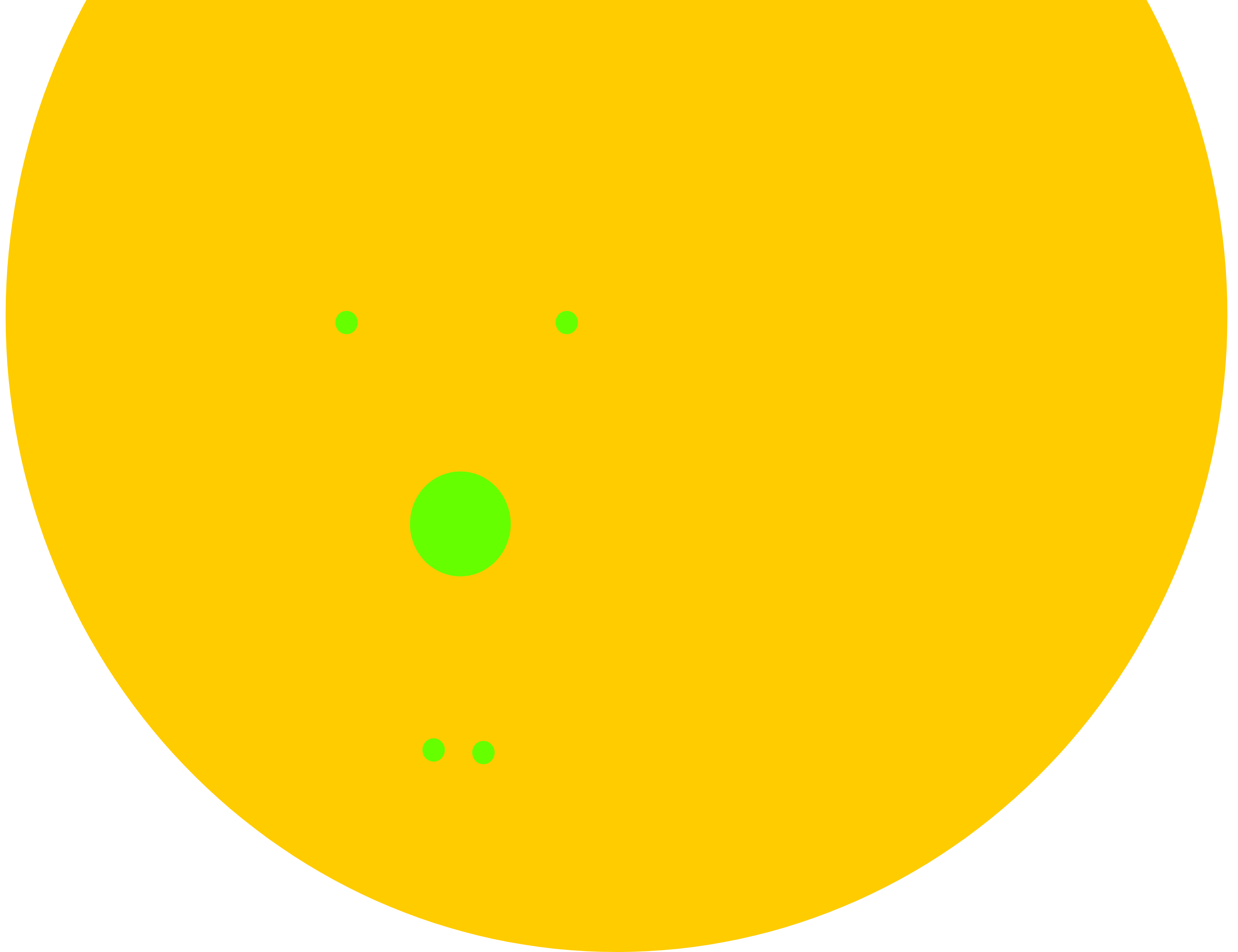

А что же так «криво»?



Все задачи которые
можно решать на C++

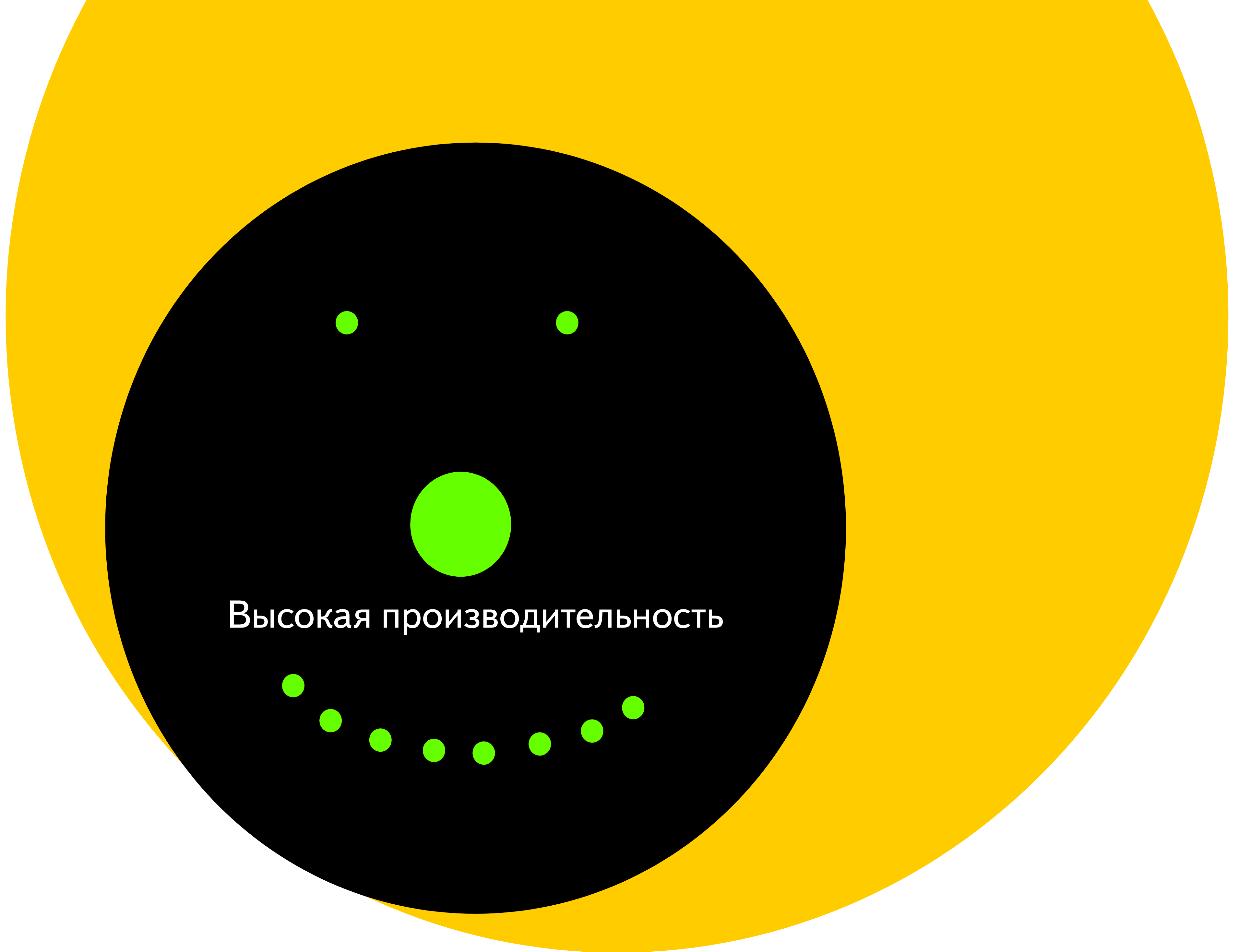




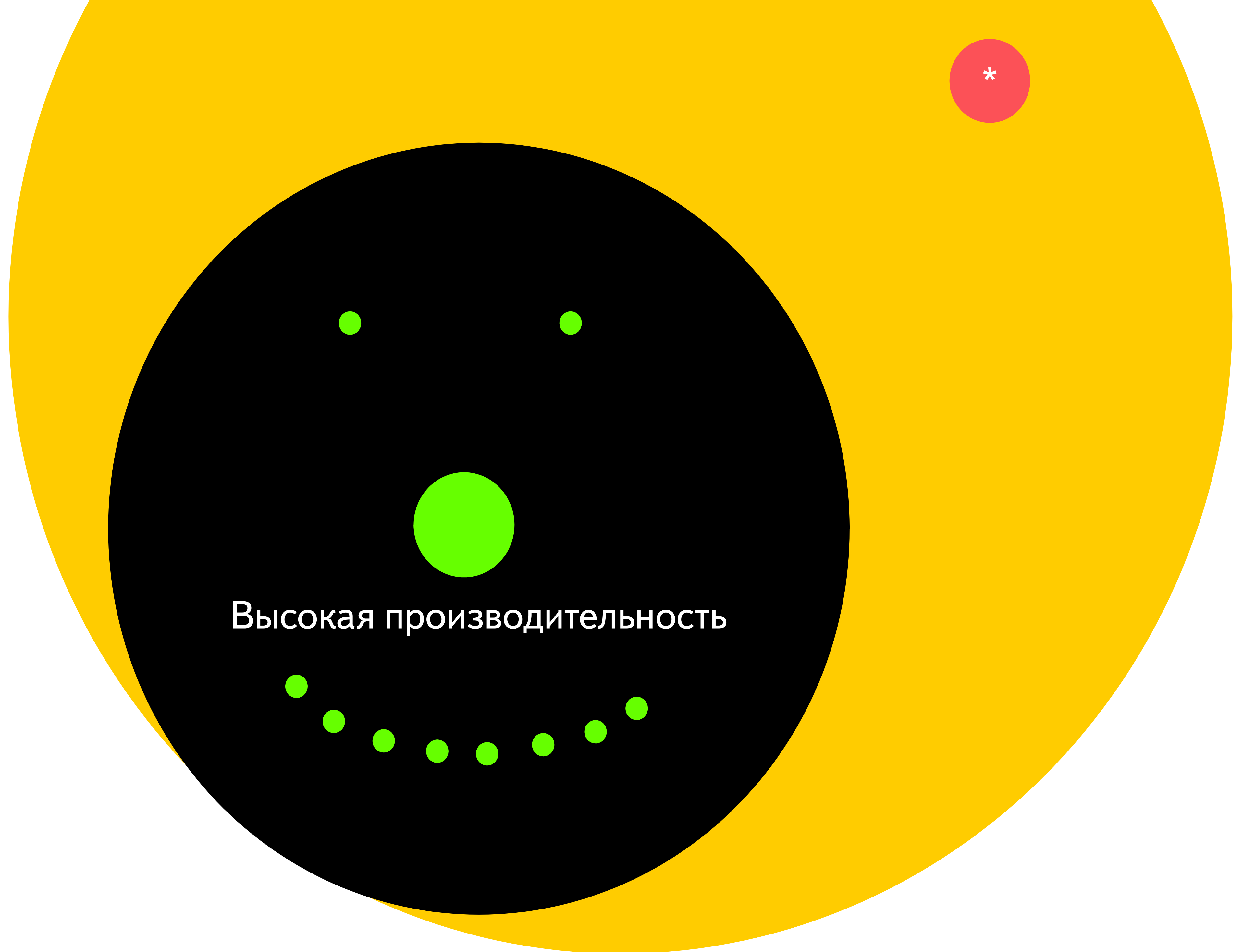




Высокая производительность



Высокая производительность



Итого:

Многие вещи в C++ не развиваются, люди в комитете — как правило эксперты по высокой нагрузке, люди работающие в крупных фирмах, с большими, уже написанными кодовыми базами.

Ява не тормозит

Немного про нашу нагрузку

(или о том как сделать больно известным решениям)

Нагрузка

1 микросервис в 1 ДЦ:

Нагрузка

1 микросервис в 1 ДЦ:
~ 20 000 событий в секунду

Нагрузка

1 микросервис в 1 ДЦ:
~ 20 000 событий в секунду
~ 30 GB логов в час

Нагрузка

1 микросервис в 1 ДЦ:
~ 20 000 событий в секунду
~ 30 GB логов в час

В сумме:

Нагрузка

1 микросервис в 1 ДЦ:
~ 20 000 событий в секунду
~ 30 GB логов в час

В сумме:
> 10^9 событий

Нагрузка

1 микросервис в 1 ДЦ:
~ 20 000 событий в секунду
~ 30 GB логов в час

В сумме:
> 10^9 событий
> 1 TB логов в час

Что с этими терабайтами данных делать?

Сохранять и анализировать :)

Logstash

– Программа для сбора, трансформации и складирования логов.

Logstash

– Программа для сбора, трансформации и складирования логов.

Бесплатное и очень популярное Open Source приложение на Java.

Logstash

– Программа для сбора, трансформации и складирования логов.

Бесплатное и очень популярное Open Source приложение на Java.

Что может пойти не так?

Logstash

– Программа для сбора, трансформации и складирования логов.

Бесплатное и очень популярное Open Source приложение на Java.

Что может пойти не так?

%CPU	%MEM	COMMAND
505,5	2,9	java
66,9	1,4	daemon

Что тормозит?

Наш pipeline

Наш pipeline

- Считываем данные с диска

Наш pipeline

- Считываем данные с диска
- Разбиваем на ключ-значение

Наш pipeline

- Считываем данные с диска
- Разбиваем на ключ-значение
- Применяем парочку простых правил

Наш pipeline

- Считываем данные с диска
- Разбиваем на ключ-значение
- Применяем парочку простых правил
- Формируем запись в новом формате

Наш pipeline

- Считываем данные с диска
- Разбиваем на ключ-значение
- Применяем парочку простых правил
- Формируем запись в новом формате
- Отсылаем в удалённое хранилище

Logstash

Вооружаемся perf...

... обнаруживаем что основная часть времени тратится в каких-то конкурентных ассоциативных контейнерах и ещё более странных местах.

Наш pipeline

- Считываем данные с диска
- Разбиваем на ключ-значение
- Применяем парочку простых правил
- Формируем запись в новом формате
- Отсылаем в удалённое хранилище

Сделаем свой logstash!

Сделаем свой logstash!

- Без лишнего

Сделаем свой logstash!

- Без лишнего
- Шустрый

Сделаем свой logstash!

- Без лишнего
- Шустрый
- Модный и современный

Пилорама

(A factory in which logs are sawed and send to remote)

Пилорама

- Считываем данные с диска
- Разбиваем на ключ-значение
- Применяем парочку простых правил
- Формируем запись в новом формате
- Отсылаем в удалённое хранилище

Пилорама

- Считываем данные с диска
- Разбиваем на ключ-значение
- Применяем парочку простых правил
- Формируем запись в новом формате
- Отсылаем в удалённое хранилище

Пилорама

- Считываем данные с диска
- Формируем запись в новом формате и с применёнными правилами
- Отсылаем в удалённое хранилище

Конвертация

```
utils::string_view chunk;
```

Конвертация

```
do {  
    const std::string key = GetKey();  
    if (state_ == State::kIncompleteRecord) {  
        // Stop parsing to let the producer write more data and finish the record.  
        return 0;  
    }  
  
    const utils::string_view value = GetValue();  
    if (state_ == State::kIncompleteRecord) { return 0; }  
    WriteWithFilters(writer, key, value);  
} while (state_ == State::kParsing);
```

Конвертация

```
do {  
    const std::string key = GetKey();  
    if (state_ == State::kIncompleteRecord) {  
        // Stop parsing to let the producer write more data and finish the record.  
        return 0;  
    }  
  
    const utils::string_view value = GetValue();  
    if (state_ == State::kIncompleteRecord) { return 0; }  
    WriteWithFilters(writer, key, value);  
} while (state_ == State::kParsing);
```


Конвертация

```
do {  
    const std::string key = GetKey();  
    if (state_ == State::kIncompleteRecord) {  
        // Stop parsing to let the producer write more data and finish the record.  
        return 0;  
    }  
  
    const utils::string_view value = GetValue();  
    if (state_ == State::kIncompleteRecord) { return 0; }  
    WriteWithFilters(writer, key, value);  
} while (state_ == State::kParsing);
```

Конвертация

```
do {  
    const std::string key = GetKey();  
    if (state_ == State::kIncompleteRecord) {  
        // Stop parsing to let the producer write more data and finish the record.  
        return 0;  
    }  
  
    const utils::string_view value = GetValue();  
    if (state_ == State::kIncompleteRecord) { return 0; }  
    WriteWithFilters(writer, key, value);  
} while (state_ == State::kParsing);
```

Конвертация

```
do {  
    const std::string key = GetKey();  
    if (state_ == State::kIncompleteRecord) {  
        // Stop parsing to let the producer write more data and finish the record.  
        return 0;  
    }  
  
    const utils::string_view value = GetValue();  
    if (state_ == State::kIncompleteRecord) { return 0; }  
  
    WriteWithFilters(writer, key, value);  
} while (state_ == State::kParsing);
```

Конвертация

```
do {  
    const std::string key = GetKey();  
    if (state_ == State::kIncompleteRecord) {  
        // Stop parsing to let the producer write more data and finish the record.  
        return 0;  
    }  
  
    const utils::string_view value = GetValue();  
    if (state_ == State::kIncompleteRecord) { return 0; }  
    WriteWithFilters(writer, key, value);  
} while (state_ == State::kParsing);
```

Скорость

Скорость

- > 360к записей в секунду
- > 75MB в секунду

Скорость

> 360к записей в секунду

> 75MB в секунду

> 270 GB в час

Скорость

- > 360к записей в секунду
- > 75MB в секунду
- > 270 GB в час:
 - ~10 микросервисов на 1ом ядре Пилорамы
 - ~ Все логи Такси на 4х ядрах Пилорамы

А что так медленно?

(немного боли)

Эх... нам бы C++20 а не cctz

```
// Formats a std::tm using strftime(3).
```

```
void FormatTM(std::string* out, const std::string& fmt, const std::tm& tm) {  
    for (int i = 2; i != 32; i *= 2) {  
        size_t buf_size = fmt.size() * i;  
        std::vector<char> buf(buf_size);  
        if (size_t len = strftime(&buf[0], buf_size, fmt.c_str(), &tm)) {  
            out->append(&buf[0], len);  
            return;  
        }  
    }  
}
```

Эх... нам бы C++20 а не cctz

```
// Formats a std::tm using strftime(3).
```

```
void FormatTM(std::string* out, const std::string& fmt, const std::tm& tm) {  
    for (int i = 2; i != 32; i *= 2) {  
        size_t buf_size = fmt.size() * i;  
        std::vector<char> buf(buf_size);  
        if (size_t len = strftime(&buf[0], buf_size, fmt.c_str(), &tm)) {  
            out->append(&buf[0], len);  
            return;  
        }  
    }  
}
```

С++ на практике

- Считываем данные с диска
- Формируем запись в новом формате и с применёнными правилами
- Отсылаем в удалённое хранилище

С++ на практике

- `boost::interprocess::mapped_region`
- Формируем запись в новом формате и с применёнными правилами
- Отсылаем в удалённое хранилище

С++ на практике

- Считываем данные с диска
- Формируем запись в новом формате и с применёнными правилами
- Отсылаем в удалённое хранилище

Отсылка

```
std::string result;  
  
do {  
    AppendNewData(result);  
    if (result.size() < treshold) {  
        engine::SleepFor(1s);  
        continue;  
    }  
    SendToRemote(result);  
    result.clear();  
} while (true);
```

Отсылка

```
std::string result;  
  
do {  
    AppendNewData(result);  
    if (result.size() < treshold) {  
        engine::SleepFor(1s);  
        continue;  
    }  
    SendToRemote(result);  
    result.clear();  
} while (true);
```


Отсылка

```
std::string result;  
  
do {  
    AppendNewData(result);  
    if (result.size() < treshold) {  
        engine::SleepFor(1s);  
        continue;  
    }  
    SendToRemote(result);  
    result.clear();  
} while (true);
```

Отсылка

```
std::string result;  
  
do {  
    AppendNewData(result);  
    if (result.size() < treshold) {  
        engine::SleepFor(1s);  
        continue;  
    }  
    SendToRemote(result);  
    result.clear();  
} while (true);
```

Sleep?

(вы точно пишете высокопроизводительный сервис?)

Как-то не похоже на высоко производительную вещь!

```
std::string result;  
  
do {  
    AppendNewData(result);  
    if (result.size() < treshold) {  
        engine::SleepFor(1s);  
        continue;  
    }  
    SendToRemote(result);  
    result.clear();  
} while (true);
```

Ой

```
std::string result;
do {
    AppendNewData(result);
    if (result.size() < treshold) {
        engine::SleepFor(1s); // асинхронный метод
        continue;
    }
    SendToRemote(result); // асинхронный метод
    result.clear();
} while (true);
```

Coroutines TS

```
std::string result;  
  
do {  
    AppendNewData(result);  
    if (result.size() < treshold) {  
        co_await engine::SleepFor(1s); // асинхронный метод  
        continue;  
    }  
    co_await SendToRemote(result); // асинхронный метод  
    result.clear();  
} while (true);
```

Coroutines TS

```
std::string result;  
  
do {  
    AppendNewData(result);  
    if (result.size() < treshold) {  
        co_await engine::SleepFor(1s); // асинхронный метод  
        continue;  
    }  
    co_await SendToRemote(result); // асинхронный метод  
    result.clear();  
} while (true);
```

Coroutines TS

```
std::string result;  
  
do {  
    AppendNewData(result);  
    if (result.size() < treshold) {  
        co_await engine::SleepFor(1s); // асинхронный метод  
        continue;  
    }  
  
    co_await SendToRemote(result); // асинхронный метод  
    result.clear();  
} while (true);
```


Чего не хватает в C++

Хотелось бы для этой задачи ...

- C++20
 - Time zones
 - char8_t
 - flat_map
 - ...

char8_t

```
void do_something(unsigned char* data, int& result) {  
    result += data[0] - u8'0';  
    result += data[1] - u8'0';  
}
```

```
void do_something(char8_t* data, int& result) {  
    result += data[0] - u8'0';  
    result += data[1] - u8'0';  
}
```

char8_t

A ▾		Left: x86-64 clang (trunk) -O3 -s... ▾	Right: x86-64 clang (trunk) -O3 -s... ▾
1 -		do_something(unsigned char*, int	1 + _Z12do_somethingPDUrI: # @_Z12dc
2		movzx eax, byte ptr [rdi]	2 movzx eax, byte ptr [rdi]
3 -		mov ecx, dword ptr [rsi]	3 + add eax, dword ptr [rsi]
4 -		lea edx, [rax + rcx]	4 + movzx ecx, byte ptr [rdi + 1]
5 -		lea eax, [rax + rcx - 48]	5 + lea eax, [rcx + rax]
6 -		mov dword ptr [rsi], eax	
7 -		movzx eax, byte ptr [rdi + 1]	
8 -		lea eax, [rax + rdx]	
9		add eax, -96	6 add eax, -96
10		mov dword ptr [rsi], eax	7 mov dword ptr [rsi], eax
11		ret	8 ret

char8_t

```
void do_something(unsigned char* data, int& result) {  
    result += data[0] - u8'0';  
    result += data[1] - u8'0';  
}
```

```
void do_something(char8_t* data, int& result) {  
    result += data[0] - u8'0';  
    result += data[1] - u8'0';  
}
```

char8_t

```
void do_something(unsigned char* data, int& result) {  
    result += data[0] - u8'0';  
    result += data[1] - u8'0';  
}
```

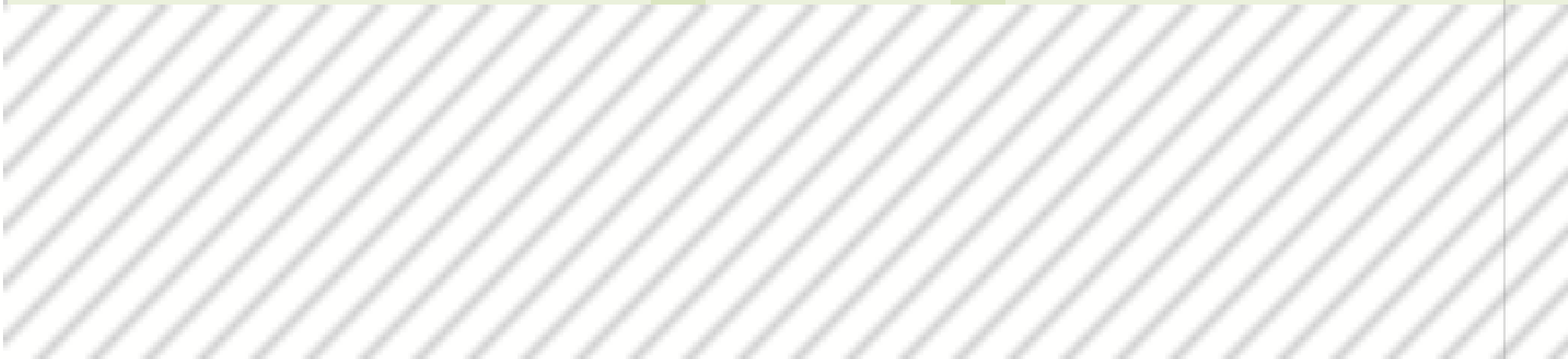
```
void do_something(char8_t* data, int& result) {  
    result += data[0] - u8'0';  
    result += data[1] - u8'0';  
}
```

char8_t

```
void do_something(unsigned char* data, int& result) {  
    result += data[0] - u8'0';  
    result += data[1] - u8'0';  
}
```

```
void do_something(char8_t* data, int& result) {  
    result += data[0] - u8'0';  
    result += data[1] - u8'0';  
}
```

char8_t

A ▾		Left: x86-64 clang (trunk) -O3 -s... ▾	Right: x86-64 clang (trunk) -O3 -s... ▾
1 -		do_something(unsigned char*, int	1 + _Z12do_somethingPDUrI: # @_Z12dc
2		movzx eax, byte ptr [rdi]	2 movzx eax, byte ptr [rdi]
3 -		mov ecx, dword ptr [rsi]	3 + add eax, dword ptr [rsi]
4 -		lea edx, [rax + rcx]	4 + movzx ecx, byte ptr [rdi + 1]
5 -		lea eax, [rax + rcx - 48]	5 + lea eax, [rcx + rax]
6 -		mov dword ptr [rsi], eax	
7 -		movzx eax, byte ptr [rdi + 1]	
8 -		lea eax, [rax + rdx]	
9		add eax, -96	6 add eax, -96
10		mov dword ptr [rsi], eax	7 mov dword ptr [rsi], eax
11		ret	8 ret

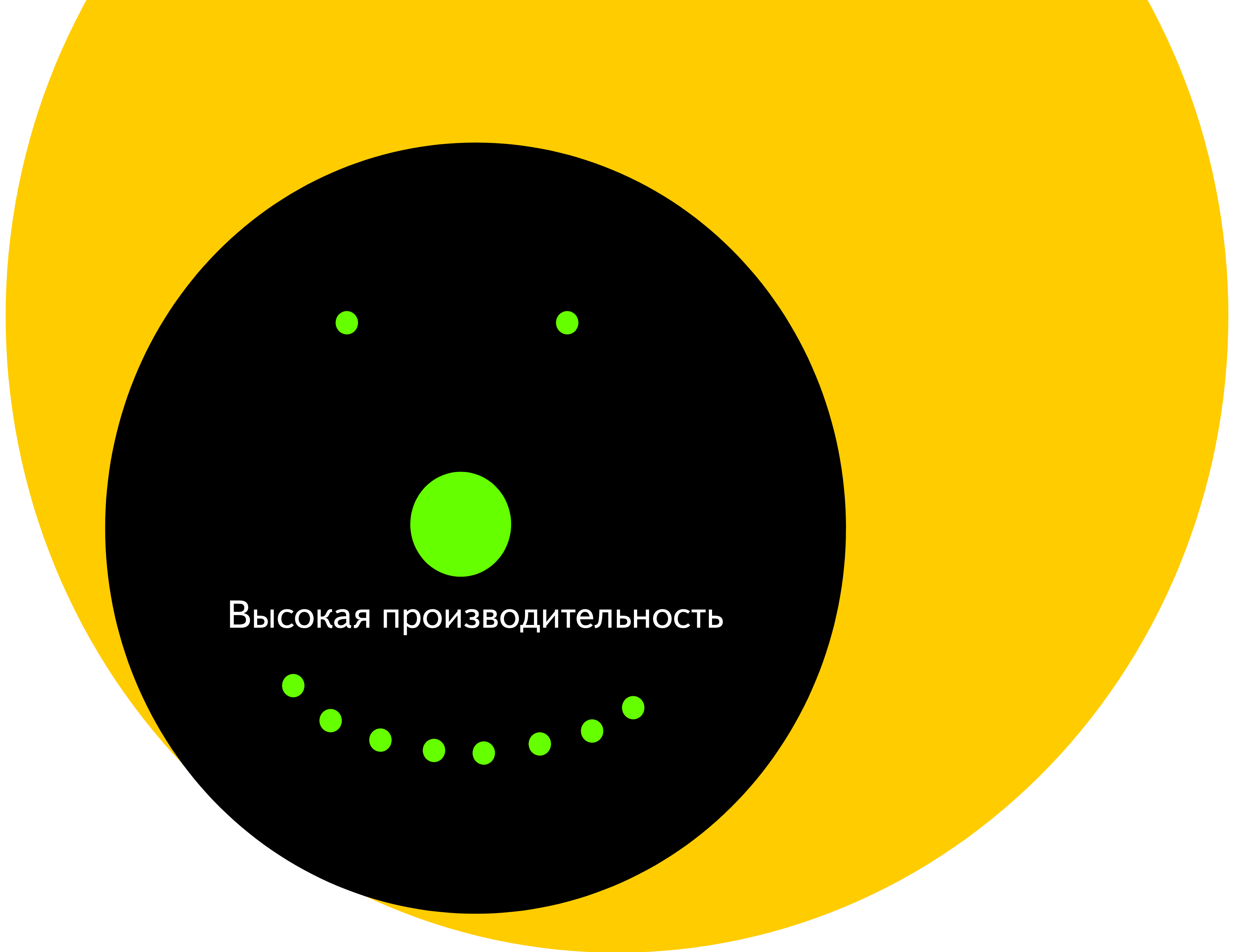
Хотелось бы для этой задачи ...

- C++20
 - Time zones
 - char8_t
 - flat_map
 - ...
- Стандартную библиотеку C++ побольше
 - mmap
 - JSON
 - Protobuf
 - ...

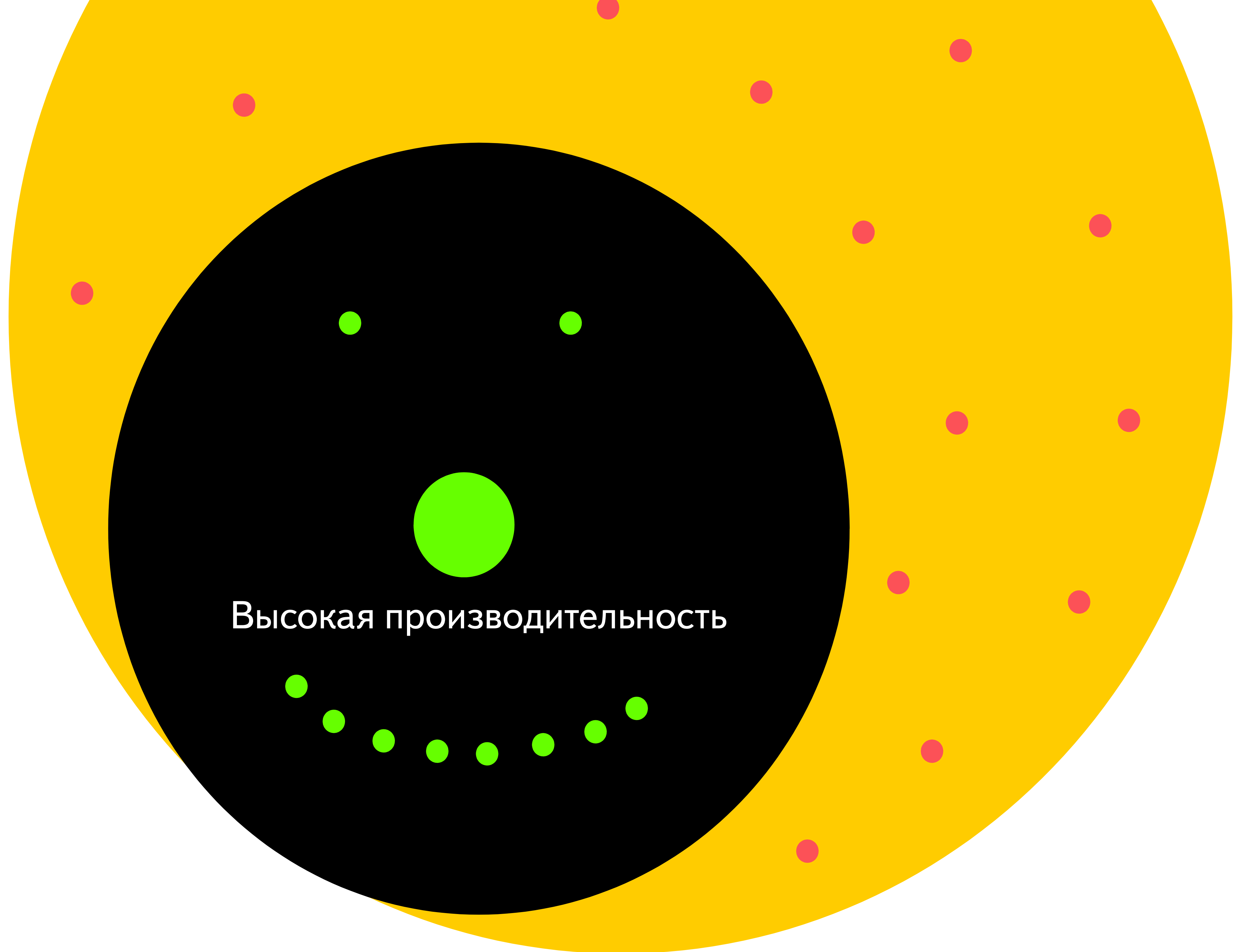
Хотелось бы для этой задачи ...

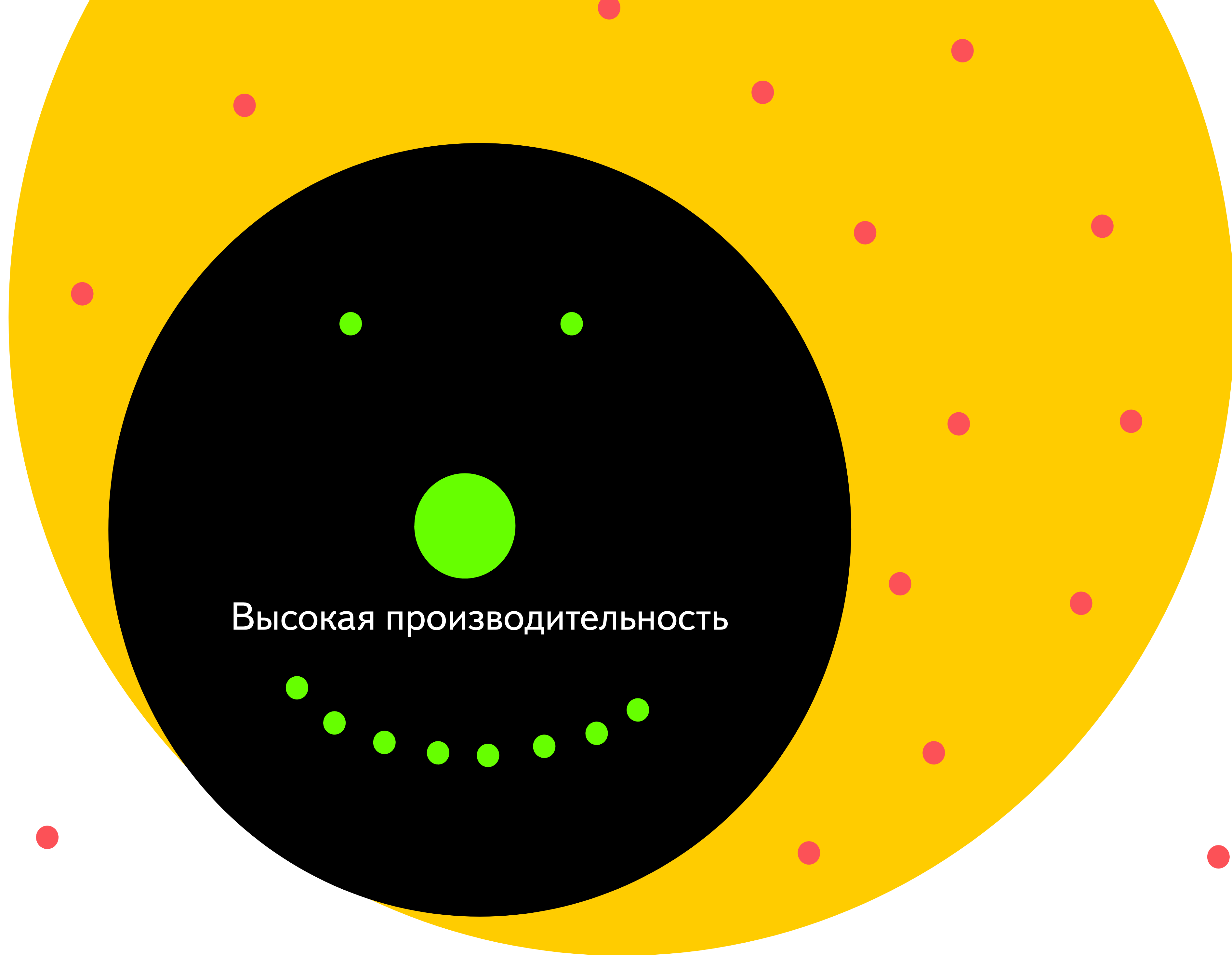
- C++20
 - Time zones
 - char8_t
 - flat_map
 - ...
- Стандартную библиотеку C++ побольше
 - mmap
 - JSON
 - Protobuf
 - ...
- Правильный Async IO
 - ASIO
 - Beast
 - AFIO
 - ...

ОК, а что делать то?



Высокая производительность





Высокая производительность

ОК, а начать то с чего?

Спасибо

Полухин Антон

Старший разработчик Yandex.Taxi



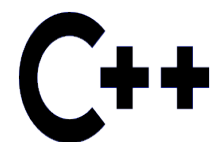
antoshkka@gmail.com



antoshkka@yandex-team.ru



<https://github.com/apolukhin>



РГ21 C++ РОССИЯ

<https://stdcpp.ru/>

