



СРАВНЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ КОМПИЛИРУЕМЫХ И ИНТЕРПРЕТИРУЕМЫХ СРЕД ПРОГРАММИРОВАНИЯ ПРИ РЕШЕНИИ ОТДЕЛЬНЫХ ЗАДАЧ ЕГЭ ПО ИНФОРМАТИКЕ

Фомин Андрей Томасович ,
учитель информатики и
физики МБОУ гимназии
№14 им. Ю А Гагарина г.
Ейска МО Ейский район

Стратегия выполнения заданий КЕГЭ

Заключается в выборе
таких инструментов,
которые бы позволяли
решать задания
максимально эффективно,
за приемлемые временные
рамки



Общие характеристики системы программирования C++

- Лидер индустрии. Любой крупный проект написан, в том числе, и на C++.
- Универсальный язык со статической типизацией и поддержкой всех существующих парадигм программирования



Общие характеристики системы программирования C++

- Позволяет писать очень быстрые приложения для всех существующих операционных систем.
- C++ имеет очень богатую стандартную библиотеку
- Есть свободно распространяемые компиляторы



Общие характеристики системы программирования Python

- Интерпретируемый язык с динамической типизацией
- Поддерживает множество парадигм программирования
- Компиляторы языка созданы для всех ведущих платформ
- Распространяется свободно



«Плюсы» Python

- Очень простой и лаконичный синтаксис («принцип наименьшего удивления»)
- Стандартная библиотека включает большой набор функций
- Наличие интерактивного режима превращает Python в мощный калькулятор



МИФЫ СВЯЗАННЫЕ С PYTHON

- Высокая скорость обучения
- Очень компактный код, по сравнению с другими языками (не для новичка!)



МИФЫ СВЯЗАННЫЕ С C++

- Сначала нужно выучить С
- Это объектно-ориентированный язык
- Эффективность достигается низкоуровневым кодом
- Подходит только для больших и сложных программ



ВАРИАНТЫ РЕШЕНИЯ ЗАДАЧИ НА PYTHON

Постановка задачи:

Дана непустая строка S и целое число N (> 0). Необходимо вывести строку, содержащую символы строки S, между которыми вставлено по N символов ch

Вариант 1. Используем пустую строку как накопитель и функции для работы со строкой

Вариант 2. Используем в качестве накопителя стандартный динамический массив List

ВАРИАНТ 1

```
S = input('S => ')
N = int(input('N => '))
ch = input('ch => ')
ch = N * ch
k = len(S)
S1 = str(S)
temp = ''
for j in range(k):
    temp = temp + S1[j] + ch
S1 = temp[:-len(ch)]
```

ВАРИАНТ 2

```
S2 = str(S)
L = list(S2)
for j in range(1, 2 * k - 1, 2):
    L.insert(j, ch)
S2 = ''.join(L)
```

Работает в 10 раз
быстрее!

ЗАДАНИЕ 16

Вычисление рекуррентных выражений

Необходимо знать:

- Как программируются рекурсивные алгоритмы
- Синтаксис условной инструкции
- Синтаксис функций

ЗАДАНИЕ 16

Определите, сколько символов * выведет эта процедура при вызове F(35):

```
def F( n ):  
    print('*')  
    if n >= 1:  
        print('*')  
        F(n-1)  
        F(n-2)  
    print('*')
```



ЗАДАНИЕ 16

```
def F( n ):  
    global i  
    i += 1 # print('*')  
    if n >= 1:  
        i += 1 # print('*')  
        F(n-1)  
        F(n-2)  
        i += 1 # print('*')
```

i = 0

F(35)

print(i)

В тех местах программы, где выводились символы «*» работает счетчик

Ответ: 96'631'265

ЗАДАНИЕ 16

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

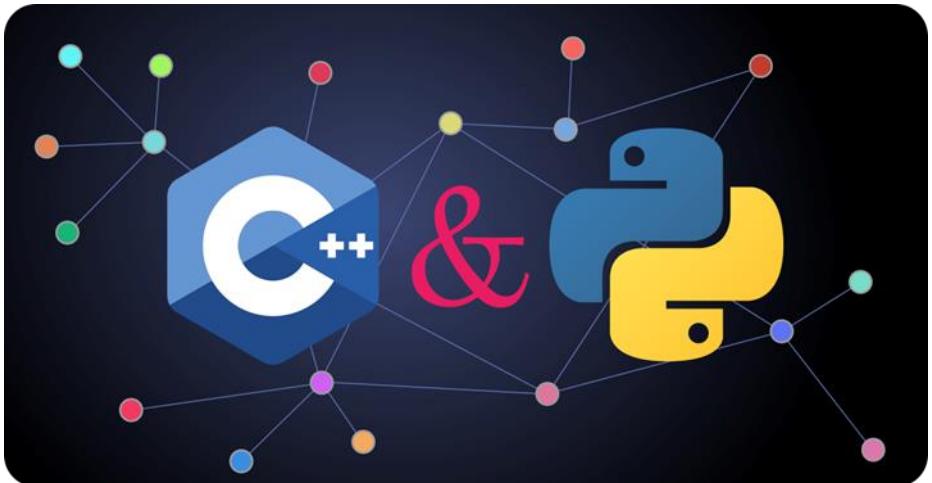
```
void F(int n, int &i) {
    i++;
    if (n >= 1) {
        i++;
        F(n-1, i);
        F(n-2, i);
        i++;
    }
}
```

```
int main() {
    int i = 0;
    F(35, i);
    cout << i << endl;
    return 0;
}
```



РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Python	C++
10.6	0.18



Более чем 50 кратное
превосходство!

ЗАДАНИЕ 16

Алгоритм вычисления функций $F(n)$ и $G(n)$ задан следующими соотношениями:

$$F(n) = G(n) = 1 \text{ при } n = 1$$

$$F(n) = F(n-1) - n \cdot G(n-1), \text{ при } n > 1$$

$$G(n) = F(n-1) + 2 \cdot G(n-1), \text{ при } n > 1$$

Чему равно значение функции $G(18)$?

ЗАДАНИЕ 16

```
def F(n):
    if n == 1:
        return 1
    elif n > 1:
        return F(n-1) - n * G(n - 1)

def G(n):
    if n == 1:
        return 1
    elif n > 1:
        return F(n - 1) + 2 * G(n - 1)

print(G(30))
```



ЗАДАНИЕ 16

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

unsigned long long G(int);
unsigned long long F(int);

int main() {
    cout << G(30) << endl;
    return 0;
}
```



ЗАДАНИЕ 16

```
unsigned long long F(int n) {  
    if (n == 1)  
        return 1;  
    else  
        if (n > 1)  
            return F(n-1) - n * G(n - 1);  
}
```



ЗАДАНИЕ 16

```
unsigned long long G(int n) {  
    if (n == 1)  
        return 1;  
    else  
        if (n > 1)  
            return F(n - 1) + 2 * G(n -  
1);  
}
```



ТЕСТИРОВАНИЕ ПРОГРАММЫ

Аргументы	Python	C++
18	0,03	0,0006
20	0,12	0,003
25	4,0	0,1
30	127,3	2,68



ЗАДАНИЕ 25

Умение создавать
собственные программы
(10–20 строк) для обработки
целочисленной информации

Для решения задачи
применяются классические
алгоритмы

Необходимо знать:

- Синтаксис циклов while, for
- Синтаксис условной инструкции if
- Синтаксис функций

ЗАДАНИЕ 25

Постановка задачи:

Напишите программу, которая ищет среди целых чисел, принадлежащих числовому отрезку $[1'820'348; 2'880'927]$, числа, имеющие ровно 5 различных делителей. Выведите эти делители для каждого найденного числа в порядке возрастания.

ЗАДАНИЕ 25



```
from math import sqrt  
cnt = 5  
start = 1203848  
end = 2880927
```

```
def Divs(n):  
    D = []  
    q = round(sqrt(n))  
    if q * q == n:  
        D = [q]  
    for d in range(1, q):  
        if n % d == 0:  
            D = D + [d, n // d]  
        if len(D) > cnt:  
            break  
    return D
```

ЗАДАНИЕ 25

```
for i in range(start, end+1):
    L = Divs(i)
    if len(L) == cnt:
        L.sort()
        for r in L:
            print(r, end=' ')
    print()
```



ЗАДАНИЕ 25



```
#include <iostream>
#include <vector>
#include <cmath>
#include <algorithm>

using namespace std;
```

```
vector<int> Divs(int n) {
    vector<int> D;
    int q = int(trunc(sqrt(n)));
    if (q * q == n)
        D.push_back(q);
    for(int d = 1; d < q; d++) {
        if (n % d == 0) {
            D.emplace_back(d);
            D.emplace_back(n / d);
            if (D.size() > 5)
                break;
        }
    }
    return D; }
```

ЗАДАНИЕ 25

```
int main() {  
    vector<int> L;  
    for(int i = 1203848; i <= 2880927; i++){  
        L = Divs(i);  
        if (L.size() == 5) {  
            sort(L.begin(), L.end());  
            for(auto r : L)  
                cout << r << " ";  
            cout << "\n";  
        }  
    }  
    return 0; }
```



РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Python	C++
41.1	3.7



ЗАДАНИЕ 25

Постановка задачи:

Рассматриваются целые числа, принадлежащих числовому отрезку $[153'732; 225'674]$, которые представляют собой произведение двух различных простых делителей. В ответе запишите количество таких чисел и такое из них, простые делители которого отличаются друг от друга меньше всего. Если чисел с наименьшей разностью делителей несколько, запишите в ответе наименьшее из них.

ЗАДАНИЕ 25

```
start = 153732
end = 225674
def Del(n):
    L = []
    for d in range(2, n // 2 + 1):
        if n % d == 0:
            L = L + [d]
        if len(L) > 2:
            break
    return L
```



Не оптимизировано!

ЗАДАНИЕ 25

```
Min = 1000000
d = None
i = 0
for j in range(start, end+1):
    D = Del(j)
    if len(D) == 2:
        i += 1
        if max(D) - min(D) < Min:
            Min = max(D) - min(D)
            d = j
print(i, d)
```



ЗАДАНИЕ 25

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

vector<int> Del(int n) {
    vector<int> L;
    for(int j = 2; j <= n / 2; j++) {
        if (n % j == 0) {
            L.push_back(j);
            if (L.size() > 2)
                break;
        }
    }
    return L; }
```



Не оптимизировано!



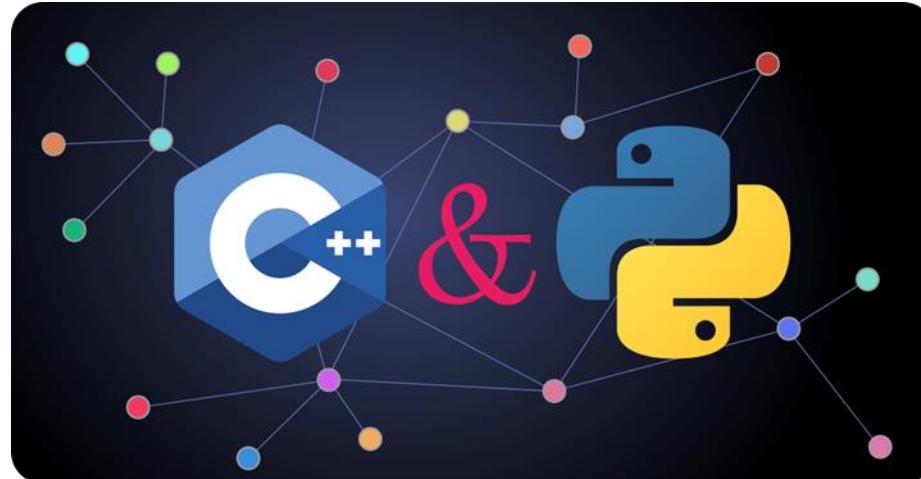
ЗАДАНИЕ 25

```
int main() {
    int begin = 153732, finish = 225674;
    int Min = 1000000;
    int d, i = 0;
    vector<int> D;
    for(int j = begin; j <= finish; j++){
        D = Del(j);
        if (D.size() == 2) {
            i++;
            if (D[1] - D[0] < Min) {
                Min = D[1] - D[0];
                d = j;
            }
        }
    }
}
```

```
cout << i << " " << d << endl;
return 0;
}
```

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

Python	C++
157,2	8,1



АЛГОРИТМ «РЕШЕТО ЭРАТОСФЕНА»

```
def sieve(n):
    pr = [True for i in range(n + 1)]
    pr[0] = pr[1] = False
    i = 2
    while i * i <= n:
        if pr[i]:
            j = i * i
            while j <= n:
                pr[j] = False
                j += i
        i += 1
    return pr
```



АЛГОРИТМ «РЕШЕТО ЭРАТОСФЕНА»

```
prime = sieve(n)
for i in range(len(prime)):
    if prime[i]:
        print(i, end=' ')
```



АЛГОРИТМ «РЕШЕТО ЭРАТОСФЕНА»

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

void sieve(long long n);
vector<bool> prime;

int main() {
    long long k;
    cout << "Введите границу для поиска = ";
    cin >> k;
```



Основная
программа

АЛГОРИТМ «РЕШЕТО ЭРАТОСФЕНА»



```
sieve(k);  
    for (size_t i = 2; i < prime.size(); i++)  
        if (prime[i])  
            cout << i << " ";  
    return 0;  
}
```

Основная
программа



АЛГОРИТМ «РЕШЕТО ЭРАТОСФЕНА»

```
void sieve(long long n) {  
    for (long long i = 0; i <= n; i++)  
        prime.push_back(true);  
    prime[0] = prime[1] = false;  
    for (long long i = 2; i * i <= n; ++i)  
        if (prime[i])  
            for (long long j = i * i; j <= n; j += i)  
                prime[j] = false;  
}
```

Реализация

ТЕСТИРОВАНИЕ ПРОГРАММЫ

Аргументы	Python	C++
100	0.001	4.7773e-05
1000	0.02	0.0009
10000	0.17	0.018
100000	0.76	0.03
1000000	5.8	0.18

На последнем teste в 30 раз быстрее!



Контакты



sakhaustro@mail.ru



<http://inf-w.ru/>