Overview of Repetition

MENYHÁRT László Gábor
(ORCID: 0000-0002-1574-4454)
ELTE Eötvös Loránd University, Budapest, Hungary
Faculty of Informatics, 3in Research Group, Martonvásár, Hungary

Abstract. There are differences between the everyday concept of repetition, the usage in informatics and the different implementations. This article collects the different levels, paradigms and language options and it presents a performed measurement and finally it evaluates the results didactically. So, it is useful for students preparing for a programming competition and their informatics teachers.

Keywords: repetition, recursion, iteration, programming languages, benchmark

1. Introduction

In the ordinary sense, repetition is a natural phenomenon: Think of regular alternations of days, weeks, seasons, etc. Nevertheless, it appears in IT in several ways. We can think differently, they occur differently in paradigms, they have different syntax and implementation in programming languages, so they run is different, so their run time can be very different. These observations and impressions have in-spired me to gather current opportunities and compare them.

This article reviews the conceptual definitions, measurement of running different implementations and didactic examination of the results. The presented method and the current results can be useful for students preparing for a programming competition and their informatics teachers.

2. The concept of repetition

Repetition is completion or execution of something several times. I collected some definition from different places, even though everyone knows or feels what it means.

Hungarian Synonym dictionary (Magyar Szinonima kéziszótár) contains the words: again (ismét) and repeat (ismétel):

„***ismét*** *újra, megint | szintén, ismételten*” [1/140. o.]

„***ismétel*** *mondogat, hajtogat, csépel| szajkóz | megismétel, elismétel, visszamond| reprodukál, megrepetál | próbál, gyakorol | folytat*” [1/140. o.]

In New Hungarian Dictionary (Új Magyar Lexikon) the definition of repetition is this: „***ismétlés*** *:* ***1.*** *(nev) oktatási eljárás ; … Osztály~: …* ***2.*** *(nyelvt) … A magyarban jelölheti a cselekmény hosszan tartó voltát (pl. ment-ment) …* ***3.*** *a stilisztikában …*” [2/448. o.]

The Hungarian ethnographic lexicon (Magyar néprajzi lexikon) [3] provides only a linguistic definition to emphasize something to say as a poetic expression.

„***ismétlés*** *(fn.) … Cselekvés, melynél fogva valamit ismét, azaz újra, még egyszer teszünk. …*” can be found in [4].

Sulinet knowledgebase at Informatics subject contains definition of repetition as ‘multiple process’ („Többszöri végrehajtás”). [5]

In WikiSzótár [6] one of the definitions of repetition is 'Redoing an action when we do the same thing we did before.' („**2.** Egy **cselekvés újra végzése**, amikor ugyanazt tesszük, amit korábban már megtettünk.").

In Cambridge dictionary **repeat** is defined as „*to happen, or to do something, more than once*" [7], while **iteration** is defined as „*the process of doing something again and again, …"* [8].

3. Repetition on levels of informatics

When we have a specific problem to solve, we just know that something will have to be done several times. At the next abstract level of our thinking, we decide whether an operation that we would like to perform several times will be a function and it calls itself, which is recursion [10], or will be repeated cyclically. This distinction is already present as a paradigm, since functional programming provides only the former.

Running of loops can be distinguished as conditional and specified number of times.

There are two groups at this specified number of times execution. First has a loop variable as a counter which changes at repetition, while the other has a given variable with the item from the series. With the first version to get the items from a series is possible with a counter which will be an index of the elements, the second version gives back immediately a given element from the series to process.

Conditional loops can be divided into two groups, first version checks the conditional at first, second checks it at the end. The difference is that the statements will be processed at least once when the checking is at last, while when the checking is at the beginning it possible that the statements are not processed. Both versions' condition is a logical expression which defines whether the statements must be processed (again) or not. Loops can be grouped by happening based on the condition. Namely whether the execution will happen or not at true expression. So, it can define the condition of staying inside or exit.

<table>
<tr><th colspan="6">Repetition</th></tr>
<tr><td rowspan="4">Recursion</td><td colspan="5">Iteration</td></tr>
<tr><td colspan="4">Conditional</td><td colspan="2">Running a certain number of items</td></tr>
<tr><td colspan="2">Check first</td><td colspan="2">Check last</td><td rowspan="2">Counter</td><td rowspan="2">Iteration on elements of series</td></tr>
<tr><td>staying inside</td><td>exit cond.</td><td>staying inside</td><td>exit cond.</td></tr>
</table>

Table 1.: Overview table of repetition

Syntax can modify this clear picture, because most programming languages do not support any of the above options. For example, exit condition must be defined in Pascal for the conditional loop where the condition will be checked at last, but in C-like languages staying inside condition must be defined. Usually staying inside condition must be defined in those conditional loops where the condition will be checked at first, but shell script in Linux there is syntax for defining exit condition. Indeed, all four types of condition must be defined. In C the `for` keyword can be used as loop which checks the condition at first, and there are `break` and `continue` keywords to end earlier the `for` loop which runs theoretically a certain number of items. Of course, with multiple nested `for` loops you have to use tricks to, so you should avoid this option. There is programming language

(like Logo), where loop-variable is not present generally, it can be used only if it required (in REPEAT the variable REPCOUNT).

Not only keywords and syntax can be different in high-level programming languages, but the compiler can change the actually running byte code and its implementation. So, their runtime can be different. I have already implemented the solution of the same tasks in different languages [11, 12, 13, 14, 15, 16] and the runtime was diverse, it occurred to me to make a more accurate measurement.

4. Description of measurement

I wrote code for five-six different loop implementation in six programming languages. Then I collected the measured run times in milliseconds.

Programming languages were C++, C#, Java, JavaScript, Pascal and Python.

I implemented recursion, loops with checking conditional at first, checking it at last, iteration with loop variables as counter and as item if the language supported. It was skipped in case of not supported. Sometimes iteration with items can be implemented in more ways, then these were merged into a group.

4.1. Presentation of the environment

The measurement ran on a computer with Intel® Core™ i7-8750H 2.20GHz processor and 32GB memory, which has 64 bit Windows 10 Pro operating system.

The C++ compiler was mingw32-g++.exe with version 5.1.0 which is default in the Code::Blocks 17.12.

I used .NER Framework's compiler for C# found on Windows 10, it is „Microsoft (R) Visual C# Compiler version 4.8.3752.0".

Java compiler was the currently most recent OpenJDK published on 6th of October 2019.

```
>java -version
openjdk version "13.0.1" 2019-10-15
OpenJDK Runtime Environment (build 13.0.1+9)
OpenJDK 64-Bit Server VM (build 13.0.1+9, mixed mode, sharing)
```

JavaScript codes ran on Node v10.16.0.

„Free Pascal Compiler version 3.0.4 [2017/10/06] for i386" was used for Pascal 32 bit.

Interpreter for Python codes was „Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)]".

4.2. Presentation of implemented algorithm

Repetition was implemented in all six programming languages and possible syntaxes based on the next algorithm.

As found, number of steps is limited in recursion in most programming languages, at first the maximum number of steps was determined. Accurate values are in paragraph 4.3 in Table 2. This

maximum value is not enough to measure all methods because it is too fast. So, codes must be run multiple times, (`for`) loop was chosen for this.

Elapsed time was calculated from the difference of start and end timestamp. Empty loop, namely loop without statements are not useable, so a simple assignment was present. But I was interested only in elapsed time of the loop, so I had to remove the time of the assignment. So, I implemented the same with two assignments as well. Abstract algorithm:

```
time1
loop
  assignment1
time2
loop
  assignment1
  assignment2
time3
```

I calculated the total time devoted to an assigment (1) and I assumed that time of assignments are equals (2). I calculated (3) the specific time of the given loop:

$$T_{a2} = (t_3 - t_2) - (t_2 - t_1) \quad (1)$$

$$T_{a1} = T_{a2} \quad (2)$$

$$T_l = t_2 - t_1 - T_{a1} \quad (3)$$

Implemented codes can be found in Appendix A.

4.3. Results of measurement

At first the maximum number of steps was determined based on the implementation of recursion which is good in all programming languages. The recursive function calls happen at the beginning of the function to avoid the optimization of tail-recursion.

Programming language	**MaxCnt (In this environment, with these program codes)**
C++	56988
C#	15918
Java	11420
JavaScript	8940
Pascal	65134
Python	4194

Table 2.: Maximum number of recursive function calls

Finally, 4000 repetition happened 10.000 times in an iteration with simple for syntax to get measurable and comparable results. Later it was detected that it was good choice because this is the fastest and that is why it does not cause confusion in comparison. So double times 40.000.000 loops were measured.

4.4. Evaluation of measurement

Appendix B. contains raw data of measurement from which average and relative deviation was determined with Excel. The time (T_i) in milliseconds concern to the first 40 million loops. Deviation is given in relative to average in percentage, so it is easier to understand and compare. Next table contains the results:

Programing language	While	For	ForEach [+lambda]	ForEach +named fnc	Recursion
C++	7 (127%)	6 (113,9%)	1142 (8,8%)	1141 (12,7%)	135 (17,7%)
C#	20 (46,8%)	24 (55,1%)	117 (43%)	113 (69,2%)	90 (10,2%)
Java	54 (42,2%)	50 (21,6%)	234 (22%)* 170 (18,7%)	2020 (12,4%)	109 (15,9%)
JavaScript	36 (37,5%)	41 (68,1%)	254 (18,6%)	242 (14,7%)	267 (9,2%)
Pascal	65 (10,1%)	89 (36,5%)	565 (8,6)*	notSupported	138 (11,3%)
Python	2054 (28,8%)	1156 (55,3%)	460 (95,4%)*	3589 (20,8%)	9876 (15,4%)

Table 3.: Results of measurement: average (deviation relative to the average in %)

4.5. Opinions of the results

You can see in the table that the specified number of steps with loop variable (for) and the conditional loop with checking at first (`while`) are the fastest repetition methods only there is deviation in Python. Recursion is the next which is slower double times in Pascal and Java, four-five times slower in C#, JavaScript and Python, and twenty times slower in C++, as the averages of previous two. Previous average times are fraction of a second but in Python it is around 10 seconds. Similar time was measured for iteration over the items of series with C# and JavaScript to the time of recursion calls, but it is nine times slower in C++, and two times or eighteen times slower in Java depends on usage of anonymous or named functions at iteration. While it is "only" three times slower in Python compared to for loop, but it is faster than recursion. For loop in C++ is the fastest, and recursion is Python is the slowest.

5. Didactic consideration

If I had to choose a programming language for education based on the table above, I would choose C#, because the measured times are consistent. Pascal does not support every method, so it should be skipped. Python is too slow. So, my order would be C#, JavaScript, C++ and Java based on the average runtime and its deviation.

Of course, all methods must be taught to the students and should be shown how they can choose between paradigms and syntaxes.

There are cases where the readability of the code is more important, but sometimes fast runtime is necessary. For instance, it was determined [9], that recursion calls usually less effective then iterations, but we often get a simpler, easier-to-read code. Such a comparative table can also be useful in programming competitions where time limits are used, and the competitor can choose

the language. However, the statement is not only an assignment, and other language elements can affect the total runtime.

The knowledge and measurement results from this article can help you to choose the right implementation for a given task. The described technique can be helpful later if Benchmark [17] would be performed again because of processor development and appearance of newer compilers may change the times described here. Because newer compiler versions can analyse the codes and create more efficient binary codes with varying changes of codes. Like Python has version 3.8, and there are alternate implementations [18, 19] as well, which would be worth comparing.

The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).

References

1. Bék Gerzson, Csifféry Tamás: *Magyar szinonima kéziszótár*, Könyvmíves Könyvkiadó, Budapest (2004)
2. Akadémiai kiadó szerkesztősége: *Új magyar lexikon*, 3 G-J, Nyolcadik, változatlan lenyomat, Akadémiai Kiadó, Budapest (1962)
3. *Magyar néprajzi lexikon*, Akadémiai Kiadó, Budapest (1977-1982)
4. Czuczor Gergely, Fogarasi János: *A magyar nyelv szótára*, Emich Gusztáv at Hungarian academic printer Pest (1862) https://czuczor.oszk.hu/kereses.php?kereses=ism%C3%A9tl%C3%A9s (last viewed: 2019.11.11.) and https://mek.oszk.hu/05800/05887/pdf/3kotet_1.pdf (page 74., last viewed: 2019.11.11.)
5. *The concept of repetition in the Sulinet knowledgebase* (2016) https://tudasbazis.sulinet.hu/hu/informatika/informatika/informatika-2-evfolyam/algoritmusok-a-hetkoznapokban-tevekenysegek-elemekre-bontasa-helyes-sorrend-megkeresese/ismetles-szerepe (last viewed: 2019.11.11.)
6. *WikiSzótár* Dictionary (2012) https://wikiszotar.hu/ertelmezo-szotar/Ism%C3%A9tl%C3%A9s (last viewed: 2019.11.11.)
7. *Repeat in Cambridge Dictionary* (2019) https://dictionary.cambridge.org/dictionary/english/repeat (last viewed: 2019.11.11.)
8. *Iteration in Cambridge Dictionary* (2019) https://dictionary.cambridge.org/dictionary/english/iteration (last viewed: 2019.11.11.)
9. Rónyai L., Ivanyos G., Szabó R.: *Algoritmusok*, Typotex (2004) page 37.
10. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest.: *Introduction to Algorithms*, MIT Press (1990) (Hungarian version: Algoritmusok, Műszaki Könyvkiadó 1997)
11. *C++ reference* (2019) https://en.cppreference.com/w/ (last viewed: 2019.11.11.)
12. *C# reference* (2017) https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/ (last viewed: 2019.11.11.)

13. *Java documentation* (2019) https://docs.oracle.com/javase/tutorial/java/ (last viewed: 2019.11.11.)

14. *JavaScript reference* (2019) https://developer.mozilla.org/hu/docs/Web/JavaScript/Reference (last viewed: 2019.11.11.)

15. *Free Pascal reference guide* (2017) https://www.freepascal.org/docs-html/ref/ref.html (last viewed: 2019.11.11.)

16. *The Python Language Reference* (2019) https://docs.python.org/2.7/reference/index.html (last viewed: 2019.11.11.)

17. *Meaning of Beenhmark* https://dictionary.cambridge.org/dictionary/english/benchmark (last viewed: 2019.12.26.)

18. *Alternative Python Implementations* https://www.python.org/download/alternatives/ (last viewed: 2019.11.11.)

19. *Cython* https://cython.org/ (last viewed: 2019.11.11.)

Appendix

A. Source codes

A.1. C++

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <array>
#include <chrono>

using namespace std;

int a;
int b;

void recursiveFor(int &i, const int &N, std::array<int,65135> &v) {
    if (i<N) {
        i++;
        recursiveFor(i,N,v);
        a=v[i-1];
    }
}

void recursiveFor2(int &i, const int &N, std::array<int,65135> &v) {
    if (i<N) {
        i++;
        recursiveFor2(i,N,v);
        a=v[i-1];
        b=v[i-1];
    }
}

int main()
{
```

```
    auto timer_start= std::chrono::high_resolution_clock::now();
    auto timer_mid= std::chrono::high_resolution_clock::now();
    auto timer_end= std::chrono::high_resolution_clock::now();
    std::chrono::duration<double> elapsed;
    std::chrono::duration<double> elapsed2;

    const int MaxN=65135;
    int N=4000;
    int times=10000;

    std::array<int,MaxN> v;
    for (int i=0; i<N; i++)
    {
        v[i];
    }

    auto func=[](const int&x)
    {
        a=x;
    };

     auto func2=[](const int&x)
    {
        a=x;
        b=x;
    };

    timer_start = std::chrono::high_resolution_clock::now();
    for (int j=0; j<times; j++)
    {
        int i=0;
        while (i<N)
        {
            a=v[i];
            i++;
        }
    }
    timer_mid = std::chrono::high_resolution_clock::now();
    for (int j=0; j<times; j++)
    {
        int i=0;
        while (i<N)
        {
            a=v[i];
            b=v[i];
            i++;
        }
    }
    timer_end = std::chrono::high_resolution_clock::now();
    elapsed = timer_mid - timer_start;
    elapsed2 = timer_end - timer_mid;
    std::cout << "Cpp;while;cnt:"<< times*N <<";lt:" << elapsed.count()*1000-
(elapsed2.count()*1000-elapsed.count()*1000) << "\n";

    timer_start = std::chrono::high_resolution_clock::now();
    for (int j=0; j<times; j++)
    {
        int i=0;
        do
        {
            a=v[i];
            i++;
        } while (i<=N);
    }
```

```
    timer_mid = std::chrono::high_resolution_clock::now();
    for (int j=0; j<times; j++)
    {
        int i=0;
        do
        {
            a=v[i];
            b=v[i];
            i++;
        } while (i<=N);
    }
    timer_end = std::chrono::high_resolution_clock::now();
    elapsed = timer_mid - timer_start;
    elapsed2 = timer_end - timer_mid;
    std::cout << "Cpp;do-while;cnt:"<< times*N <<";lt:" << elapsed.count()*1000-
(elapsed2.count()*1000-elapsed.count()*1000) << "\n";

    timer_start = std::chrono::high_resolution_clock::now();
    for (int j=0; j<times; j++)
    {
        for (int i=0; i<N; i++)
        {
            a=v[i];
        }
    }
    timer_mid = std::chrono::high_resolution_clock::now();
    for (int j=0; j<times; j++)
    {
        for (int i=0; i<N; i++)
        {
            a=v[i];
            b=v[i];
        }
    }
    timer_end = std::chrono::high_resolution_clock::now();
    elapsed = timer_mid - timer_start;
    elapsed2 = timer_end - timer_mid;
    std::cout << "Cpp;for;cnt:"<< times*N <<";lt:" << elapsed.count()*1000-
(elapsed2.count()*1000-elapsed.count()*1000) << "\n";

    timer_start = std::chrono::high_resolution_clock::now();
    for (int j=0; j<times; j++)
    {
        std::for_each(v.begin(), v.end(),
                      [](const int&x)
        {
            a=x;
        });
    }
    timer_mid = std::chrono::high_resolution_clock::now();
    for (int j=0; j<times; j++)
    {
        std::for_each(v.begin(), v.end(),
                      [](const int&x)
        {
            a=x;
            b=x;
        });
    }
    timer_end = std::chrono::high_resolution_clock::now();
    elapsed = timer_mid - timer_start;
    elapsed2 = timer_end - timer_mid;
    std::cout << "Cpp;ForEach + lambda;cnt:"<< times*N <<";lt:" <<
elapsed.count()*1000-(elapsed2.count()*1000-elapsed.count()*1000) << "\n";
```

```
    timer_start = std::chrono::high_resolution_clock::now();
    for (int j=0; j<times; j++)
    {
        std::for_each(v.begin(), v.end(), func);
    }
    timer_mid = std::chrono::high_resolution_clock::now();
    for (int j=0; j<times; j++)
    {
        std::for_each(v.begin(), v.end(), func2);
    }
    timer_end = std::chrono::high_resolution_clock::now();
    elapsed = timer_mid - timer_start;
    elapsed2 = timer_end - timer_mid;
    std::cout << "Cpp;ForEach + named function;cnt:"<< times*N <<";lt:" <<
elapsed.count()*1000-(elapsed2.count()*1000-elapsed.count()*1000) << "\n";

    timer_start = std::chrono::high_resolution_clock::now();
    for (int j=0; j<times; j++)
    {
        int i=0;
        recursiveFor(i,N,v);
    }
    timer_mid = std::chrono::high_resolution_clock::now();
    for (int j=0; j<times; j++)
    {
        int i=0;
        recursiveFor2(i,N,v);
    }
    timer_end = std::chrono::high_resolution_clock::now();
    elapsed = timer_mid - timer_start;
    elapsed2 = timer_end - timer_mid;
    std::cout << "Cpp;Recursive;cnt:"<< times*N <<";lt:" << elapsed.count()*1000-
(elapsed2.count()*1000-elapsed.count()*1000) << "\n";

    return 0;
}
```

A.2. C#

```
using System;

public class Program
{
    static int a;
    static int b;

    static void recursiveFor(int i, int N, int[] v) {
        if (i<N) {
            i++;
            recursiveFor(i,N,v);
            a=v[i-1];
        }
    }

    static void recursiveFor2(int i, int N, int[] v) {
        if (i<N) {
            i++;
            recursiveFor2(i,N,v);
            a=v[i-1];
            b=v[i-1];
        }
```

```
    }

    public static void Main()
    {
    DateTime timer_start= DateTime.Now;
    DateTime timer_mid= DateTime.Now;
    DateTime timer_end;
    TimeSpan elapsed;
    TimeSpan elapsed2;

    int N=4000;
    int times=10000;

    int[] v=new int[N];
    for (int i=0; i<N; i++)
    {
        v[i]=i;
    }

    Action<int> func = x => {
        a=x;
    };
    Action<int> func2 = x => {
        a=x;
        b=x;
    };

    timer_start= DateTime.Now;
    for (int j=0; j<times; j++)
    {
        for (int i=0; i<N; i++)
        {
            a=v[i];
        }
    }
    timer_mid= DateTime.Now;
    for (int j=0; j<times; j++)
    {
        for (int i=0; i<N; i++)
        {
            a=v[i];
            b=v[i];
        }
    }
    timer_end = DateTime.Now;
    elapsed = timer_mid - timer_start;
    elapsed2 = timer_end - timer_mid;
    Console.WriteLine("cs;for;cnt:{1};lt:{0}",elapsed.Milliseconds-
(elapsed2.Milliseconds-elapsed.Milliseconds),N*times);

    timer_start= DateTime.Now;
    for (int j=0; j<times; j++)
    {
        int i=0;
        while (i<N)
        {
            a=v[i];
            i++;
        }
    }
    timer_mid= DateTime.Now;
    for (int j=0; j<times; j++)
    {
        int i=0;
```

```
            while ( i<N)
            {
                a=v[i];
                b=v[i];
                i++;
            }
        }
        timer_end = DateTime.Now;
        elapsed = timer_mid - timer_start;
        elapsed2 = timer_end - timer_mid;
        Console.WriteLine("cs;while;cnt:{1};lt:{0}",elapsed.Milliseconds-
(elapsed2.Milliseconds-elapsed.Milliseconds),N*times);

        timer_start= DateTime.Now;
        for (int j=0; j<times; j++)
        {
            Array.ForEach(v, (int x) =>
            {
                a=x;
            });
        }
        timer_mid= DateTime.Now;
        for (int j=0; j<times; j++)
        {
            Array.ForEach(v, (int x) =>
            {
                a=x;
                b=x;
            });
        }
        timer_end = DateTime.Now;
        elapsed = timer_mid - timer_start;
        elapsed2 = timer_end - timer_mid;
        Console.WriteLine("cs;ForEach + lambda;cnt:{1};lt:{0}",elapsed.Milliseconds-
(elapsed2.Milliseconds-elapsed.Milliseconds),N*times);

        timer_start= DateTime.Now;
        for (int j=0; j<times; j++)
        {
            Array.ForEach(v,func);
        }
        timer_mid= DateTime.Now;
        for (int j=0; j<times; j++)
        {
            Array.ForEach(v,func2);
        }
        timer_end = DateTime.Now;
        elapsed = timer_mid - timer_start;
        elapsed2 = timer_end - timer_mid;
        Console.WriteLine("cs;ForEach + named
function;cnt:{1};lt:{0}",elapsed.Milliseconds-(elapsed2.Milliseconds-
elapsed.Milliseconds),N*times);

        timer_start= DateTime.Now;
        for (int j=0; j<times; j++)
        {
            int i=0;
            recursiveFor(i,N,v);
        }
        timer_mid= DateTime.Now;
        for (int j=0; j<times; j++)
        {
            int i=0;
            recursiveFor2(i,N,v);
```

```
    }
    timer_end = DateTime.Now;
    elapsed = timer_mid - timer_start;
    elapsed2 = timer_end - timer_mid;
    Console.WriteLine("cs;Recursive;cnt:{1};lt:{0}",elapsed.Milliseconds-
(elapsed2.Milliseconds-elapsed.Milliseconds),N*times);
}
}
```

A.3. Java

```
import java.util.Date;
import java.util.List;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.function.Consumer;

public class ciklus{

    static Integer a;
    static Integer b;

    public static void recursiveFor(int i, int N, Integer[] v) {
        if (i<N) {
            i++;
            recursiveFor(i,N,v);
            a=v[i-1];
        }
    }

    public static void recursiveFor2(int i, int N, Integer[] v) {
        if (i<N) {
            i++;
            recursiveFor2(i,N,v);
            a=v[i-1];
            b=v[i-1];
        }
    }

    public static void main(String[] args) {
        Date timer_start=new Date();
        Date timer_mid=new Date();
        Date timer_end;
        long elapsed;
        long elapsed2;

        int MaxN=65135;//100000000;
        int N=4000;//100000000;
        int times=10000; //30;

    Integer[] v=new Integer[MaxN];
        for (int i=0; i<N; i++)
        {
            v[i]=i;
        }

        Consumer<Integer> func = (x) ->
        {
            a=x;
        };
        Consumer<Integer> func2 = (x) ->
        {
```

```
            a=x;
            b=x;
        };

    timer_start = new Date();
    for (int j=0; j<times; j++)
    {
        for (int i=0; i<N; i++)
        {
            a=v[i];
        }
    }
    timer_mid = new Date();
    for (int j=0; j<times; j++)
    {
        for (int i=0; i<N; i++)
        {
            a=v[i];
            b=v[i];
        }
    }
    timer_end = new Date();
    elapsed = timer_mid.getTime() - timer_start.getTime();
    elapsed2 = timer_end.getTime() - timer_mid.getTime();
    System.out.println("Java;for;cnt:"+times*N+";lt:"+(elapsed-(elapsed2-
elapsed)));

    timer_start = new Date();
    for (int j=0; j<times; j++)
    {
        int i=0;
        while (i<N)
        {
            a=v[i];
            i++;
        }
    }
    timer_mid = new Date();
    for (int j=0; j<times; j++)
    {
        int i=0;
        while (i<N)
        {
            a=v[i];
            b=v[i];
            i++;
        }
    }
    timer_end = new Date();
    elapsed = timer_mid.getTime() - timer_start.getTime();
    elapsed2 = timer_end.getTime() - timer_mid.getTime();
    System.out.println("Java;while;cnt:"+times*N+";lt:"+(elapsed-(elapsed2-
elapsed)));

    timer_start = new Date();
    for (int j=0; j<times; j++)
    {
        for (Integer x : v)
        {
            a=x;
        }
    }
    timer_mid = new Date();
    for (int j=0; j<times; j++)
```

```
    {
        for (Integer x : v)
        {
            a=x;
            b=x;
        }
    }
    timer_end = new Date();
    elapsed = timer_mid.getTime() - timer_start.getTime();
    elapsed2 = timer_end.getTime() - timer_mid.getTime();
    System.out.println("Java;forEach;cnt:"+times*N+";lt:"+(elapsed-(elapsed2-
elapsed)));

    List<Integer> vl=Arrays.asList(v);
    timer_start =new Date();
    for (int j=0; j<times; j++)
    {
        vl.forEach((x)-> {
            a=x;
        });
    }
    timer_mid =new Date();
    for (int j=0; j<times; j++)
    {
        vl.forEach(x-> {
            a=x;
            b=x;
        });
    }
    timer_end = new Date();
    elapsed = timer_mid.getTime() - timer_start.getTime();
    elapsed2 = timer_end.getTime() - timer_mid.getTime();
    System.out.println("Java;ForEach + lambda;cnt:"+times*N+";lt:"+(elapsed-
(elapsed2-elapsed)));

    timer_start =new Date();
    for (int j=0; j<times; j++)
    {
        vl.forEach(func);
    }
       timer_mid =new Date();
    for (int j=0; j<times; j++)
    {
        vl.forEach(func2);
    }
    timer_end = new Date();
    elapsed = timer_mid.getTime() - timer_start.getTime();
    elapsed2 = timer_end.getTime() - timer_mid.getTime();
    System.out.println("Java;ForEach + named
function;cnt:"+times*N+";lt:"+(elapsed-(elapsed2-elapsed)));

    timer_start = new Date();
    try {
    for (int j=0; j<times; j++)
    {
        int i=0;
        recursiveFor(i,N,v);
    }
    } catch(Exception ex) {
    }
    timer_mid = new Date();
    try {
    for (int j=0; j<times; j++)
    {
```

```
            int i=0;
            recursiveFor2(i,N,v);
        }
        } catch(Exception ex) {
        }
        timer_end = new Date();
        elapsed = timer_mid.getTime() - timer_start.getTime();
        elapsed2 = timer_end.getTime() - timer_mid.getTime();
        System.out.println("Java;Recursive;cnt:"+times*N+";lt:"+(elapsed-(elapsed2-
elapsed)));
        }
}
```

A.4. JavaScript

```
let N=4000;
let times=10000;
let a;
let b;
let v=[];
for (let i=0; i<N; i++)
{
    v.push(i);
}

function recursiveFor(i, N, v) {
    if (i<N) {
        i++;
        recursiveFor(i,N,v);
        a=v[i-1];
    }
}

function recursiveFor2(i, N, v) {
    if (i<N) {
        i++;
        recursiveFor2(i,N,v);
        a=v[i-1];
        b=v[i-1];
    }
}

var timer_start = Date.now();
for (let j=0; j<times; j++)
{
    for (let i=0; i<N; i++)
    {
        a=i;
    }
}
var timer_mid = Date.now();
for (let j=0; j<times; j++)
{
    for (let i=0; i<N; i++)
    {
        a=i;
        b=i;
    }
}
var timer_end = Date.now();
elapsed=timer_mid-timer_start;
elapsed2=timer_end-timer_mid;
```

```
console.log("Js;for;cnt:",(times*N),";lt:",elapsed-(elapsed2-elapsed));

var timer_start = Date.now();
for (let j=0; j<times; j++)
{
    let i=0;
    while (i<N)
    {
        a=i;
        i++;
    }
}
var timer_mid = Date.now();
for (let j=0; j<times; j++)
{
    let i=0;
    while (i<N)
    {
        a=i;
        b=i;
        i++;
    }
}
var timer_end = Date.now();
elapsed=timer_mid-timer_start;
elapsed2=timer_end-timer_mid;
console.log("Js;While;cnt:",(times*N),";lt:",elapsed-(elapsed2-elapsed));

var timer_start = Date.now();
for (let j=0; j<times; j++)
{
    v.forEach((item, index, arr)=>{
        a=item;
    });
}
var timer_mid = Date.now();
for (let j=0; j<times; j++)
{
    v.forEach((item, index, arr)=>{
        a=item;
        b=item;
    });
}
var timer_end = Date.now();
elapsed=timer_mid-timer_start;
elapsed2=timer_end-timer_mid;
console.log("Js;ForEach + lambda function;cnt:",(times*N),";lt:",elapsed-
(elapsed2-elapsed));

function func(item, index, arr) {
  a=item;
}
function func2(item, index, arr) {
  a=item;
  b=item;
}
var timer_start = Date.now();
for (let j=0; j<times; j++)
{
    v.forEach(func);
}
var timer_mid = Date.now();
for (let j=0; j<times; j++)
{
```

```
    v.forEach(func2);
}
var timer_end = Date.now();
elapsed=timer_mid-timer_start;
elapsed2=timer_end-timer_mid;
console.log("Js;ForEach + named function;cnt:",(times*N),";lt:",elapsed-
(elapsed2-elapsed));

var timer_start = Date.now();
for (let j=0; j<times; j++)
{
    let i=0;
    recursiveFor(i,N,v);
}
var timer_mid = Date.now();
for (let j=0; j<times; j++)
{
    let i=0;
    recursiveFor2(i,N,v);
}
var timer_end = Date.now();
elapsed=timer_mid-timer_start;
elapsed2=timer_end-timer_mid;
console.log("Js;Recursive;cnt:",(times*N),";lt:",elapsed-(elapsed2-elapsed));
```

A.5. Pascal

```
Program ciklus;

Uses sysutils;

  var
    timer_start:TDateTime;
    timer_mid:TDateTime;
    timer_end:TDateTime;
    elapsed:TDateTime;
    elapsed2:TDateTime;
    N:integer;
    times:integer;
    v:array[1..65135] of integer;
    i:integer;
    j:integer;
    a:integer;
    b:integer;
    w:integer;

procedure recursiveFor(var i:integer; const N:integer; var v:array of integer);
begin
    if (i<=N) then begin
        i:=i+1;
        recursiveFor(i,N,v);
        a:=v[i-1];
    end
end;

procedure recursiveFor2(var i:integer; const N:integer; var v:array of integer);
begin
    if (i<=N) then begin
        i:=i+1;
        recursiveFor2(i,N,v);
        a:=v[i-1];
        b:=v[i-1];
```

```
    end
end;

BEGIN
    N:=4000;
    times:=10000;

    for i:=1 to N do begin
        v[i]:=i;
    end;

    timer_start:=Now;
    for j:=1 to times do begin
        for i:=1 to N do begin
            a:=v[i];
        end;
    end;
    timer_mid:=Now;
    for j:=1 to times do begin
        for i:=1 to N do begin
            a:=v[i];
            b:=v[i];
        end;
    end;
    timer_end:=Now;
    elapsed:=timer_mid-timer_start;
    elapsed2:=timer_end-timer_mid;
    WriteLn('Pas;for;cnt:',times*N,';lt:',(elapsed-(elapsed2-
elapsed))*100000000:6:0);

    timer_start:=Now;
    for j:=1 to times do begin
        i:=1;
        while (i<=N) do begin
            a:=v[i];
            i:=i+1;
        end;
    end;
    timer_mid:=Now;
    for j:=1 to times do begin
        i:=1;
        while (i<=N) do begin
            a:=v[i];
            b:=v[i];
            i:=i+1;
        end;
    end;
    timer_end:=Now;
    elapsed:=timer_mid-timer_start;
    elapsed2:=timer_end-timer_mid;
    WriteLn('Pas;while;cnt:',times*N,';lt:',(elapsed-(elapsed2-
elapsed))*100000000:6:0);

    timer_start:=Now;
    for j:=1 to times do begin
        for w in v do begin
            a:=w;
        end;
    end;
    timer_mid:=Now;
    for j:=1 to times do begin
        for w in v do begin
            a:=w;
            b:=w;
```

```
        end;
    end;
    timer_end:=Now;
    elapsed:=timer_mid-timer_start;
    elapsed2:=timer_end-timer_mid;
    WriteLn('Pas;forEach;cnt:',times*N,';lt:',(elapsed-(elapsed2-
elapsed))*100000000:6:0);

    timer_start:=Now;
    for j:=1 to times do begin
    end;
    timer_end:=Now;
    elapsed:=timer_end-timer_start;
    WriteLn('Pas;ForEach + lambda;cnt:',times*N,';lt:Not supported');
    timer_start:=Now;
    for j:=1 to times do begin
    end;
    timer_end:=Now;
    elapsed:=timer_end-timer_start;
    WriteLn('Pas;ForEach + named function;cnt:',times*N,';lt:Not supported');
    timer_start:=Now;
    for j:=1 to times do begin
        i:=1;
        recursiveFor(i,N,v);
    end;
    timer_mid:=Now;
    for j:=1 to times do begin
        i:=1;
        recursiveFor2(i,N,v);
    end;
    timer_end:=Now;
    elapsed:=timer_mid-timer_start;
    elapsed2:=timer_end-timer_mid;
    WriteLn('Pas;Recursive;cnt:',times*N,';lt:',(elapsed-(elapsed2-
elapsed))*100000000:6:0);

END.
```

A.6. Python

```
import sys
from datetime import datetime

#print sys.getrecursionlimit()
sys.setrecursionlimit(100000)
#print sys.getrecursionlimit()

N=4000;
times=10000;
v=[]
for i in range(1,N):
  v.append(i);

def func(x):
  a=x

def func2(x):
  a=x
  b=x

def recursiveFor(i,N,v):
  if i<N-1:
    i=i+1
```

```
    recursiveFor(i,N,v);
    a=v[i-1]

def recursiveFor2(i,N,v):
  if i<N-1:
    i=i+1
    recursiveFor2(i,N,v);
    a=v[i-1]
    b=v[i-1]

timer_start = datetime.now()
for j in range(1,times):
  for i in range(1,N):
    a=v[i-1]
timer_mid = datetime.now()
for j in range(1,times):
  for i in range(1,N):
    a=v[i-1]
    b=v[i-1]
timer_end = datetime.now()
elapsed=timer_mid-timer_start
elapsed2=timer_end-timer_mid
print
'Py;For;cnt:',N*times,';lt:',(elapsed.seconds*1000+elapsed.microseconds/1000)-
((elapsed2.seconds*1000+elapsed2.microseconds/1000)-
(elapsed.seconds*1000+elapsed.microseconds/1000))

timer_start = datetime.now()
for j in range(1,times):
  i=0
  while i<N-1:
    #std::cout << v[i] << std::endl;
    a=v[i]
    i=i+1
timer_mid = datetime.now()
for j in range(1,times):
  i=0
  while i<N-1:
    a=v[i]
    b=v[i]
    i=i+1
timer_end = datetime.now()
elapsed=timer_mid-timer_start
elapsed2=timer_end-timer_mid
print
'Py;While;cnt:',N*times,';lt:',(elapsed.seconds*1000+elapsed.microseconds/1000)-
((elapsed2.seconds*1000+elapsed2.microseconds/1000)-
(elapsed.seconds*1000+elapsed.microseconds/1000))

timer_start = datetime.now()
for j in range(1,times):
  for x in v:
    a=x
timer_mid = datetime.now()
for j in range(1,times):
  for x in v:
    a=x
    b=x
timer_end = datetime.now()
elapsed=timer_mid-timer_start
elapsed2=timer_end-timer_mid
print 'Py;ForEach [+
lambda];cnt:',N*times,';lt:',(elapsed.seconds*1000+elapsed.microseconds/1000)-
```

```
((elapsed2.seconds*1000+elapsed2.microseconds/1000)-
(elapsed.seconds*1000+elapsed.microseconds/1000))

timer_start = datetime.now()
for j in range(1,times):
  for x in v:
    func(x)
timer_mid = datetime.now()
for j in range(1,times):
  for x in v:
    func2(x)
timer_end = datetime.now()
elapsed=timer_mid-timer_start
elapsed2=timer_end-timer_mid
print 'Py;ForEach + named
function;cnt:',N*times,';lt:',(elapsed.seconds*1000+elapsed.microseconds/1000)-
((elapsed2.seconds*1000+elapsed2.microseconds/1000)-
(elapsed.seconds*1000+elapsed.microseconds/1000))

timer_start = datetime.now()
for j in range(1,times):
  i=0
  recursiveFor(i,N,v);
timer_mid = datetime.now()
for j in range(1,times):
  i=0
  recursiveFor2(i,N,v);
timer_end = datetime.now()
elapsed=timer_mid-timer_start
elapsed2=timer_end-timer_mid
print
'Py;recursive;cnt:',N*times,';lt:',(elapsed.seconds*1000+elapsed.microseconds/100
0)-((elapsed2.seconds*1000+elapsed2.microseconds/1000)-
(elapsed.seconds*1000+elapsed.microseconds/1000))
```

B. Raw data of measurement

```
Cpp;while;cnt:40000000;lt:0.056
Cpp;do-while;cnt:40000000;lt:0.658
Cpp;for;cnt:40000000;lt:31.361
Cpp;ForEach + lambda;cnt:40000000;lt:1076.23
Cpp;ForEach + named function;cnt:40000000;lt:1087.14
Cpp;Recursive;cnt:40000000;lt:124.673
cs;for;cnt:40000000;lt:22
cs;while;cnt:40000000;lt:19
cs;ForEach + lambda;cnt:40000000;lt:92
cs;ForEach + named function;cnt:40000000;lt:89
cs;Recursive;cnt:40000000;lt:85
Java;for;cnt:40000000;lt:56
Java;while;cnt:40000000;lt:51
Java;forEach;cnt:40000000;lt:213
Java;ForEach + lambda;cnt:40000000;lt:153
Java;ForEach + named function;cnt:40000000;lt:1901
Java;Recursive;cnt:40000000;lt:98
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 34
Js;ForEach + lambda function;cnt: 40000000 ;lt: 249
Js;ForEach + named function;cnt: 40000000 ;lt: 229
Js;Recursive;cnt: 40000000 ;lt: 263
Pas;for;cnt:40000000;lt:    97
Pas;while;cnt:40000000;lt:    58
Pas;forEach;cnt:40000000;lt:   549
```

```
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   130
Py;For;cnt: 40000000 ;lt: 1140
Py;While;cnt: 40000000 ;lt: 1872
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 265
Py;ForEach + named function;cnt: 40000000 ;lt: 3176
Py;recursive;cnt: 40000000 ;lt: 9022

Cpp;while;cnt:40000000;lt:11.553
Cpp;do-while;cnt:40000000;lt:0.589
Cpp;for;cnt:40000000;lt:22.609
Cpp;ForEach + lambda;cnt:40000000;lt:1095.92
Cpp;ForEach + named function;cnt:40000000;lt:1125.63
Cpp;Recursive;cnt:40000000;lt:110.243
cs;for;cnt:40000000;lt:31
cs;while;cnt:40000000;lt:8
cs;ForEach + lambda;cnt:40000000;lt:117
cs;ForEach + named function;cnt:40000000;lt:92
cs;Recursive;cnt:40000000;lt:91
Java;for;cnt:40000000;lt:37
Java;while;cnt:40000000;lt:48
Java;forEach;cnt:40000000;lt:199
Java;ForEach + lambda;cnt:40000000;lt:157
Java;ForEach + named function;cnt:40000000;lt:1961
Java;Recursive;cnt:40000000;lt:138
Js;for;cnt: 40000000 ;lt: 37
Js;While;cnt: 40000000 ;lt: 30
Js;ForEach + lambda function;cnt: 40000000 ;lt: 235
Js;ForEach + named function;cnt: 40000000 ;lt: 227
Js;Recursive;cnt: 40000000 ;lt: 251
Pas;for;cnt:40000000;lt:   101
Pas;while;cnt:40000000;lt:    94
Pas;forEach;cnt:40000000;lt:   546
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   118
Py;For;cnt: 40000000 ;lt: 1204
Py;While;cnt: 40000000 ;lt: 2665
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 270
Py;ForEach + named function;cnt: 40000000 ;lt: 3197
Py;recursive;cnt: 40000000 ;lt: 8994

Cpp;while;cnt:40000000;lt:24.209
Cpp;do-while;cnt:40000000;lt:1.789
Cpp;for;cnt:40000000;lt:4.774
Cpp;ForEach + lambda;cnt:40000000;lt:1095
Cpp;ForEach + named function;cnt:40000000;lt:1109.37
Cpp;Recursive;cnt:40000000;lt:126.22
cs;for;cnt:40000000;lt:47
cs;while;cnt:40000000;lt:2
cs;ForEach + lambda;cnt:40000000;lt:90
cs;ForEach + named function;cnt:40000000;lt:92
cs;Recursive;cnt:40000000;lt:93
Java;for;cnt:40000000;lt:50
Java;while;cnt:40000000;lt:66
Java;forEach;cnt:40000000;lt:228
Java;ForEach + lambda;cnt:40000000;lt:157
Java;ForEach + named function;cnt:40000000;lt:1938
Java;Recursive;cnt:40000000;lt:112
Js;for;cnt: 40000000 ;lt: 37
Js;While;cnt: 40000000 ;lt: 33
Js;ForEach + lambda function;cnt: 40000000 ;lt: 238
Js;ForEach + named function;cnt: 40000000 ;lt: 231
```

```
Js;Recursive;cnt: 40000000 ;lt: 251
Pas;for;cnt:40000000;lt:    82
Pas;while;cnt:40000000;lt:    62
Pas;forEach;cnt:40000000;lt:   560
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   159
Py;For;cnt: 40000000 ;lt: 1042
Py;While;cnt: 40000000 ;lt: 2833
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 235
Py;ForEach + named function;cnt: 40000000 ;lt: 3176
Py;recursive;cnt: 40000000 ;lt: 9232

Cpp;while;cnt:40000000;lt:4.446
Cpp;do-while;cnt:40000000;lt:3.817
Cpp;for;cnt:40000000;lt:0.397
Cpp;ForEach + lambda;cnt:40000000;lt:1093.52
Cpp;ForEach + named function;cnt:40000000;lt:1108.86
Cpp;Recursive;cnt:40000000;lt:127.066
cs;for;cnt:40000000;lt:13
cs;while;cnt:40000000;lt:8
cs;ForEach + lambda;cnt:40000000;lt:125
cs;ForEach + named function;cnt:40000000;lt:82
cs;Recursive;cnt:40000000;lt:93
Java;for;cnt:40000000;lt:58
Java;while;cnt:40000000;lt:31
Java;forEach;cnt:40000000;lt:211
Java;ForEach + lambda;cnt:40000000;lt:166
Java;ForEach + named function;cnt:40000000;lt:1922
Java;Recursive;cnt:40000000;lt:113
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 34
Js;ForEach + lambda function;cnt: 40000000 ;lt: 247
Js;ForEach + named function;cnt: 40000000 ;lt: 231
Js;Recursive;cnt: 40000000 ;lt: 271
Pas;for;cnt:40000000;lt:    83
Pas;while;cnt:40000000;lt:    67
Pas;forEach;cnt:40000000;lt:   556
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   133
Py;For;cnt: 40000000 ;lt: 1131
Py;While;cnt: 40000000 ;lt: 2009
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 277
Py;ForEach + named function;cnt: 40000000 ;lt: 2980
Py;recursive;cnt: 40000000 ;lt: 7478

Cpp;while;cnt:40000000;lt:3.886
Cpp;do-while;cnt:40000000;lt:0.068
Cpp;for;cnt:40000000;lt:2.29
Cpp;ForEach + lambda;cnt:40000000;lt:1114.82
Cpp;ForEach + named function;cnt:40000000;lt:1160.19
Cpp;Recursive;cnt:40000000;lt:134.357
cs;for;cnt:40000000;lt:20
cs;while;cnt:40000000;lt:21
cs;ForEach + lambda;cnt:40000000;lt:91
cs;ForEach + named function;cnt:40000000;lt:93
cs;Recursive;cnt:40000000;lt:86
Java;for;cnt:40000000;lt:65
Java;while;cnt:40000000;lt:61
Java;forEach;cnt:40000000;lt:222
Java;ForEach + lambda;cnt:40000000;lt:160
Java;ForEach + named function;cnt:40000000;lt:2018
Java;Recursive;cnt:40000000;lt:117
```

```
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 34
Js;ForEach + lambda function;cnt: 40000000 ;lt: 252
Js;ForEach + named function;cnt: 40000000 ;lt: 239
Js;Recursive;cnt: 40000000 ;lt: 270
Pas;for;cnt:40000000;lt:    83
Pas;while;cnt:40000000;lt:    67
Pas;forEach;cnt:40000000;lt:   564
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   137
Py;For;cnt: 40000000 ;lt: 1190
Py;While;cnt: 40000000 ;lt: 1832
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 300
Py;ForEach + named function;cnt: 40000000 ;lt: 3291
Py;recursive;cnt: 40000000 ;lt: 9734

Cpp;while;cnt:40000000;lt:5.952
Cpp;do-while;cnt:40000000;lt:3.779
Cpp;for;cnt:40000000;lt:5.6
Cpp;ForEach + lambda;cnt:40000000;lt:1136.82
Cpp;ForEach + named function;cnt:40000000;lt:1141.19
Cpp;Recursive;cnt:40000000;lt:133.593
cs;for;cnt:40000000;lt:23
cs;while;cnt:40000000;lt:22
cs;ForEach + lambda;cnt:40000000;lt:93
cs;ForEach + named function;cnt:40000000;lt:91
cs;Recursive;cnt:40000000;lt:89
Java;for;cnt:40000000;lt:38
Java;while;cnt:40000000;lt:54
Java;forEach;cnt:40000000;lt:218
Java;ForEach + lambda;cnt:40000000;lt:177
Java;ForEach + named function;cnt:40000000;lt:2018
Java;Recursive;cnt:40000000;lt:110
Js;for;cnt: 40000000 ;lt: 40
Js;While;cnt: 40000000 ;lt: 40
Js;ForEach + lambda function;cnt: 40000000 ;lt: 253
Js;ForEach + named function;cnt: 40000000 ;lt: 239
Js;Recursive;cnt: 40000000 ;lt: 265
Pas;for;cnt:40000000;lt:    81
Pas;while;cnt:40000000;lt:    66
Pas;forEach;cnt:40000000;lt:   565
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   131
Py;For;cnt: 40000000 ;lt: 101
Py;While;cnt: 40000000 ;lt: 1431
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 1225
Py;ForEach + named function;cnt: 40000000 ;lt: 5895
Py;recursive;cnt: 40000000 ;lt: 14090

Cpp;while;cnt:40000000;lt:5.244
Cpp;do-while;cnt:40000000;lt:0.978
Cpp;for;cnt:40000000;lt:3.862
Cpp;ForEach + lambda;cnt:40000000;lt:1139.48
Cpp;ForEach + named function;cnt:40000000;lt:1143.29
Cpp;Recursive;cnt:40000000;lt:133.426
cs;for;cnt:40000000;lt:23
cs;while;cnt:40000000;lt:21
cs;ForEach + lambda;cnt:40000000;lt:93
cs;ForEach + named function;cnt:40000000;lt:94
cs;Recursive;cnt:40000000;lt:89
Java;for;cnt:40000000;lt:38
Java;while;cnt:40000000;lt:48
```

```
Java;forEach;cnt:40000000;lt:232
Java;ForEach + lambda;cnt:40000000;lt:172
Java;ForEach + named function;cnt:40000000;lt:2012
Java;Recursive;cnt:40000000;lt:114
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 36
Js;ForEach + lambda function;cnt: 40000000 ;lt: 247
Js;ForEach + named function;cnt: 40000000 ;lt: 237
Js;Recursive;cnt: 40000000 ;lt: 272
Pas;for;cnt:40000000;lt:    83
Pas;while;cnt:40000000;lt:    67
Pas;forEach;cnt:40000000;lt:   565
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   139
Py;For;cnt: 40000000 ;lt: 1654
Py;While;cnt: 40000000 ;lt: 2797
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 265
Py;ForEach + named function;cnt: 40000000 ;lt: 3279
Py;recursive;cnt: 40000000 ;lt: 10467

Cpp;while;cnt:40000000;lt:5.164
Cpp;do-while;cnt:40000000;lt:2.288
Cpp;for;cnt:40000000;lt:4.173
Cpp;ForEach + lambda;cnt:40000000;lt:1149.63
Cpp;ForEach + named function;cnt:40000000;lt:1146.72
Cpp;Recursive;cnt:40000000;lt:134.877
cs;for;cnt:40000000;lt:24
cs;while;cnt:40000000;lt:22
cs;ForEach + lambda;cnt:40000000;lt:93
cs;ForEach + named function;cnt:40000000;lt:91
cs;Recursive;cnt:40000000;lt:111
Java;for;cnt:40000000;lt:59
Java;while;cnt:40000000;lt:52
Java;forEach;cnt:40000000;lt:247
Java;ForEach + lambda;cnt:40000000;lt:107
Java;ForEach + named function;cnt:40000000;lt:2035
Java;Recursive;cnt:40000000;lt:114
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 34
Js;ForEach + lambda function;cnt: 40000000 ;lt: 259
Js;ForEach + named function;cnt: 40000000 ;lt: 244
Js;Recursive;cnt: 40000000 ;lt: 280
Pas;for;cnt:40000000;lt:    83
Pas;while;cnt:40000000;lt:    65
Pas;forEach;cnt:40000000;lt:   646
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   152
Py;For;cnt: 40000000 ;lt: 1313
Py;While;cnt: 40000000 ;lt: 1905
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 268
Py;ForEach + named function;cnt: 40000000 ;lt: 3216
Py;recursive;cnt: 40000000 ;lt: 9942

Cpp;while;cnt:40000000;lt:0.999
Cpp;do-while;cnt:40000000;lt:2.048
Cpp;for;cnt:40000000;lt:0.98
Cpp;ForEach + lambda;cnt:40000000;lt:1216.67
Cpp;ForEach + named function;cnt:40000000;lt:1147.93
Cpp;Recursive;cnt:40000000;lt:139.67
cs;for;cnt:40000000;lt:23
cs;while;cnt:40000000;lt:21
cs;ForEach + lambda;cnt:40000000;lt:95
```

```
cs;ForEach + named function;cnt:40000000;lt:94
cs;Recursive;cnt:40000000;lt:89
Java;for;cnt:40000000;lt:41
Java;while;cnt:40000000;lt:51
Java;forEach;cnt:40000000;lt:279
Java;ForEach + lambda;cnt:40000000;lt:193
Java;ForEach + named function;cnt:40000000;lt:1988
Java;Recursive;cnt:40000000;lt:44
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 35
Js;ForEach + lambda function;cnt: 40000000 ;lt: 241
Js;ForEach + named function;cnt: 40000000 ;lt: 228
Js;Recursive;cnt: 40000000 ;lt: 272
Pas;for;cnt:40000000;lt:    81
Pas;while;cnt:40000000;lt:    67
Pas;forEach;cnt:40000000;lt:   583
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   134
Py;For;cnt: 40000000 ;lt: 809
Py;While;cnt: 40000000 ;lt: 2704
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 234
Py;ForEach + named function;cnt: 40000000 ;lt: 3207
Py;recursive;cnt: 40000000 ;lt: 10193

Cpp;while;cnt:40000000;lt:1.675
Cpp;do-while;cnt:40000000;lt:1.435
Cpp;for;cnt:40000000;lt:4.495
Cpp;ForEach + lambda;cnt:40000000;lt:1117.01
Cpp;ForEach + named function;cnt:40000000;lt:1149.84
Cpp;Recursive;cnt:40000000;lt:127.438
cs;for;cnt:40000000;lt:21
cs;while;cnt:40000000;lt:7
cs;ForEach + lambda;cnt:40000000;lt:118
cs;ForEach + named function;cnt:40000000;lt:92
cs;Recursive;cnt:40000000;lt:86
Java;for;cnt:40000000;lt:57
Java;while;cnt:40000000;lt:47
Java;forEach;cnt:40000000;lt:228
Java;ForEach + lambda;cnt:40000000;lt:175
Java;ForEach + named function;cnt:40000000;lt:1965
Java;Recursive;cnt:40000000;lt:117
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 33
Js;ForEach + lambda function;cnt: 40000000 ;lt: 244
Js;ForEach + named function;cnt: 40000000 ;lt: 218
Js;Recursive;cnt: 40000000 ;lt: 259
Pas;for;cnt:40000000;lt:    82
Pas;while;cnt:40000000;lt:    64
Pas;forEach;cnt:40000000;lt:   544
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   132
Py;For;cnt: 40000000 ;lt: 75
Py;While;cnt: 40000000 ;lt: 3586
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 975
Py;ForEach + named function;cnt: 40000000 ;lt: 3906
Py;recursive;cnt: 40000000 ;lt: 8930

Cpp;while;cnt:40000000;lt:48.477
Cpp;do-while;cnt:40000000;lt:174.121
Cpp;for;cnt:40000000;lt:0.981
Cpp;ForEach + lambda;cnt:40000000;lt:1697.38
Cpp;ForEach + named function;cnt:40000000;lt:1959.92
```

```
Cpp;Recursive;cnt:40000000;lt:216.816
cs;for;cnt:40000000;lt:88
cs;while;cnt:40000000;lt:60
cs;ForEach + lambda;cnt:40000000;lt:311
cs;ForEach + named function;cnt:40000000;lt:212
cs;Recursive;cnt:40000000;lt:128
Java;for;cnt:40000000;lt:76
Java;while;cnt:40000000;lt:183
Java;forEach;cnt:40000000;lt:510
Java;ForEach + lambda;cnt:40000000;lt:324
Java;ForEach + named function;cnt:40000000;lt:3482
Java;Recursive;cnt:40000000;lt:148
Js;for;cnt: 40000000 ;lt: 206
Js;While;cnt: 40000000 ;lt: 115
Js;ForEach + lambda function;cnt: 40000000 ;lt: 529
Js;ForEach + named function;cnt: 40000000 ;lt: 448
Js;Recursive;cnt: 40000000 ;lt: 400
Pas;for;cnt:40000000;lt:   278
Pas;while;cnt:40000000;lt:    60
Pas;forEach;cnt:40000000;lt:   817
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   215
Py;For;cnt: 40000000 ;lt: 1964
Py;While;cnt: 40000000 ;lt: 3432
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 1442
Py;ForEach + named function;cnt: 40000000 ;lt: 5341
Py;recursive;cnt: 40000000 ;lt: 14436

Cpp;while;cnt:40000000;lt:13.586
Cpp;do-while;cnt:40000000;lt:1.62
Cpp;for;cnt:40000000;lt:5.154
Cpp;ForEach + lambda;cnt:40000000;lt:1106.2
Cpp;ForEach + named function;cnt:40000000;lt:1108.6
Cpp;Recursive;cnt:40000000;lt:127.709
cs;for;cnt:40000000;lt:14
cs;while;cnt:40000000;lt:31
cs;ForEach + lambda;cnt:40000000;lt:113
cs;ForEach + named function;cnt:40000000;lt:88
cs;Recursive;cnt:40000000;lt:94
Java;for;cnt:40000000;lt:34
Java;while;cnt:40000000;lt:40
Java;forEach;cnt:40000000;lt:211
Java;ForEach + lambda;cnt:40000000;lt:154
Java;ForEach + named function;cnt:40000000;lt:1977
Java;Recursive;cnt:40000000;lt:110
Js;for;cnt: 40000000 ;lt: 34
Js;While;cnt: 40000000 ;lt: 35
Js;ForEach + lambda function;cnt: 40000000 ;lt: 249
Js;ForEach + named function;cnt: 40000000 ;lt: 249
Js;Recursive;cnt: 40000000 ;lt: 260
Pas;for;cnt:40000000;lt:    81
Pas;while;cnt:40000000;lt:    69
Pas;forEach;cnt:40000000;lt:   515
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   128
Py;For;cnt: 40000000 ;lt: 1158
Py;While;cnt: 40000000 ;lt: 1598
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 735
Py;ForEach + named function;cnt: 40000000 ;lt: 3325
Py;recursive;cnt: 40000000 ;lt: 9003

Cpp;while;cnt:40000000;lt:7.54
```

```
Cpp;do-while;cnt:40000000;lt:6.197
Cpp;for;cnt:40000000;lt:0.663
Cpp;ForEach + lambda;cnt:40000000;lt:1136.38
Cpp;ForEach + named function;cnt:40000000;lt:1125.48
Cpp;Recursive;cnt:40000000;lt:128.11
cs;for;cnt:40000000;lt:21
cs;while;cnt:40000000;lt:19
cs;ForEach + lambda;cnt:40000000;lt:95
cs;ForEach + named function;cnt:40000000;lt:98
cs;Recursive;cnt:40000000;lt:102
Java;for;cnt:40000000;lt:62
Java;while;cnt:40000000;lt:49
Java;forEach;cnt:40000000;lt:219
Java;ForEach + lambda;cnt:40000000;lt:138
Java;ForEach + named function;cnt:40000000;lt:2032
Java;Recursive;cnt:40000000;lt:89
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 34
Js;ForEach + lambda function;cnt: 40000000 ;lt: 246
Js;ForEach + named function;cnt: 40000000 ;lt: 239
Js;Recursive;cnt: 40000000 ;lt: 255
Pas;for;cnt:40000000;lt:    83
Pas;while;cnt:40000000;lt:    52
Pas;forEach;cnt:40000000;lt:   574
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   148
Py;For;cnt: 40000000 ;lt: 1175
Py;While;cnt: 40000000 ;lt: 1874
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 233
Py;ForEach + named function;cnt: 40000000 ;lt: 3260
Py;recursive;cnt: 40000000 ;lt: 9195

Cpp;while;cnt:40000000;lt:13.119
Cpp;do-while;cnt:40000000;lt:17.027
Cpp;for;cnt:40000000;lt:9.737
Cpp;ForEach + lambda;cnt:40000000;lt:1119.74
Cpp;ForEach + named function;cnt:40000000;lt:1099.06
Cpp;Recursive;cnt:40000000;lt:128.921
cs;for;cnt:40000000;lt:21
cs;while;cnt:40000000;lt:21
cs;ForEach + lambda;cnt:40000000;lt:98
cs;ForEach + named function;cnt:40000000;lt:94
cs;Recursive;cnt:40000000;lt:83
Java;for;cnt:40000000;lt:54
Java;while;cnt:40000000;lt:42
Java;forEach;cnt:40000000;lt:220
Java;ForEach + lambda;cnt:40000000;lt:174
Java;ForEach + named function;cnt:40000000;lt:1971
Java;Recursive;cnt:40000000;lt:100
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 33
Js;ForEach + lambda function;cnt: 40000000 ;lt: 242
Js;ForEach + named function;cnt: 40000000 ;lt: 233
Js;Recursive;cnt: 40000000 ;lt: 284
Pas;for;cnt:40000000;lt:    82
Pas;while;cnt:40000000;lt:    64
Pas;forEach;cnt:40000000;lt:   549
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   134
Py;For;cnt: 40000000 ;lt: 1145
Py;While;cnt: 40000000 ;lt: 1776
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 729
```

```
Py;ForEach + named function;cnt: 40000000 ;lt: 3322
Py;recursive;cnt: 40000000 ;lt: 8969

Cpp;while;cnt:40000000;lt:3.234
Cpp;do-while;cnt:40000000;lt:0.561
Cpp;for;cnt:40000000;lt:4.021
Cpp;ForEach + lambda;cnt:40000000;lt:1119.4
Cpp;ForEach + named function;cnt:40000000;lt:1111.89
Cpp;Recursive;cnt:40000000;lt:128.048
cs;for;cnt:40000000;lt:22
cs;while;cnt:40000000;lt:19
cs;ForEach + lambda;cnt:40000000;lt:94
cs;ForEach + named function;cnt:40000000;lt:91
cs;Recursive;cnt:40000000;lt:87
Java;for;cnt:40000000;lt:53
Java;while;cnt:40000000;lt:49
Java;forEach;cnt:40000000;lt:232
Java;ForEach + lambda;cnt:40000000;lt:184
Java;ForEach + named function;cnt:40000000;lt:1972
Java;Recursive;cnt:40000000;lt:100
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 33
Js;ForEach + lambda function;cnt: 40000000 ;lt: 243
Js;ForEach + named function;cnt: 40000000 ;lt: 230
Js;Recursive;cnt: 40000000 ;lt: 260
Pas;for;cnt:40000000;lt:    79
Pas;while;cnt:40000000;lt:    65
Pas;forEach;cnt:40000000;lt:   553
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   130
Py;For;cnt: 40000000 ;lt: 1162
Py;While;cnt: 40000000 ;lt: 2469
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 248
Py;ForEach + named function;cnt: 40000000 ;lt: 5494
Py;recursive;cnt: 40000000 ;lt: 9184

Cpp;while;cnt:40000000;lt:0.22
Cpp;do-while;cnt:40000000;lt:0.463
Cpp;for;cnt:40000000;lt:5.005
Cpp;ForEach + lambda;cnt:40000000;lt:1107.88
Cpp;ForEach + named function;cnt:40000000;lt:1121.43
Cpp;Recursive;cnt:40000000;lt:131.073
cs;for;cnt:40000000;lt:21
cs;while;cnt:40000000;lt:22
cs;ForEach + lambda;cnt:40000000;lt:93
cs;ForEach + named function;cnt:40000000;lt:92
cs;Recursive;cnt:40000000;lt:93
Java;for;cnt:40000000;lt:36
Java;while;cnt:40000000;lt:53
Java;forEach;cnt:40000000;lt:225
Java;ForEach + lambda;cnt:40000000;lt:159
Java;ForEach + named function;cnt:40000000;lt:1950
Java;Recursive;cnt:40000000;lt:95
Js;for;cnt: 40000000 ;lt: 32
Js;While;cnt: 40000000 ;lt: 31
Js;ForEach + lambda function;cnt: 40000000 ;lt: 250
Js;ForEach + named function;cnt: 40000000 ;lt: 236
Js;Recursive;cnt: 40000000 ;lt: 263
Pas;for;cnt:40000000;lt:    82
Pas;while;cnt:40000000;lt:    67
Pas;forEach;cnt:40000000;lt:   546
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
```

```
Pas;Recursive;cnt:40000000;lt:   127
Py;For;cnt: 40000000 ;lt: 1197
Py;While;cnt: 40000000 ;lt: 1660
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 47
Py;ForEach + named function;cnt: 40000000 ;lt: 2791
Py;recursive;cnt: 40000000 ;lt: 8820

Cpp;while;cnt:40000000;lt:0.233
Cpp;do-while;cnt:40000000;lt:2.049
Cpp;for;cnt:40000000;lt:6.047
Cpp;ForEach + lambda;cnt:40000000;lt:1114.01
Cpp;ForEach + named function;cnt:40000000;lt:1118.2
Cpp;Recursive;cnt:40000000;lt:133.1
cs;for;cnt:40000000;lt:29
cs;while;cnt:40000000;lt:23
cs;ForEach + lambda;cnt:40000000;lt:92
cs;ForEach + named function;cnt:40000000;lt:92
cs;Recursive;cnt:40000000;lt:86
Java;for;cnt:40000000;lt:55
Java;while;cnt:40000000;lt:52
Java;forEach;cnt:40000000;lt:224
Java;ForEach + lambda;cnt:40000000;lt:161
Java;ForEach + named function;cnt:40000000;lt:1966
Java;Recursive;cnt:40000000;lt:117
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 35
Js;ForEach + lambda function;cnt: 40000000 ;lt: 244
Js;ForEach + named function;cnt: 40000000 ;lt: 235
Js;Recursive;cnt: 40000000 ;lt: 261
Pas;for;cnt:40000000;lt:    79
Pas;while;cnt:40000000;lt:    64
Pas;forEach;cnt:40000000;lt:   550
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   137
Py;For;cnt: 40000000 ;lt: 81
Py;While;cnt: 40000000 ;lt: 2897
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 216
Py;ForEach + named function;cnt: 40000000 ;lt: 3273
Py;recursive;cnt: 40000000 ;lt: 11002

Cpp;while;cnt:40000000;lt:1.006
Cpp;do-while;cnt:40000000;lt:0.084
Cpp;for;cnt:40000000;lt:2.409
Cpp;ForEach + lambda;cnt:40000000;lt:1114.54
Cpp;ForEach + named function;cnt:40000000;lt:1099.17
Cpp;Recursive;cnt:40000000;lt:128.804
cs;for;cnt:40000000;lt:22
cs;while;cnt:40000000;lt:19
cs;ForEach + lambda;cnt:40000000;lt:94
cs;ForEach + named function;cnt:40000000;lt:89
cs;Recursive;cnt:40000000;lt:86
Java;for;cnt:40000000;lt:54
Java;while;cnt:40000000;lt:51
Java;forEach;cnt:40000000;lt:209
Java;ForEach + lambda;cnt:40000000;lt:160
Java;ForEach + named function;cnt:40000000;lt:1966
Java;Recursive;cnt:40000000;lt:113
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 35
Js;ForEach + lambda function;cnt: 40000000 ;lt: 241
Js;ForEach + named function;cnt: 40000000 ;lt: 234
Js;Recursive;cnt: 40000000 ;lt: 257
Pas;for;cnt:40000000;lt:    79
```

```
Pas;while;cnt:40000000;lt:    67
Pas;forEach;cnt:40000000;lt:   545
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   134
Py;For;cnt: 40000000 ;lt: 461
Py;While;cnt: 40000000 ;lt: 2933
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 1159
Py;ForEach + named function;cnt: 40000000 ;lt: 3164
Py;recursive;cnt: 40000000 ;lt: 11564

Cpp;while;cnt:40000000;lt:0.98
Cpp;do-while;cnt:40000000;lt:2.854
Cpp;for;cnt:40000000;lt:6.992
Cpp;ForEach + lambda;cnt:40000000;lt:1116.53
Cpp;ForEach + named function;cnt:40000000;lt:1115.34
Cpp;Recursive;cnt:40000000;lt:127.015
cs;for;cnt:40000000;lt:38
cs;while;cnt:40000000;lt:8
cs;ForEach + lambda;cnt:40000000;lt:102
cs;ForEach + named function;cnt:40000000;lt:93
cs;Recursive;cnt:40000000;lt:80
Java;for;cnt:40000000;lt:64
Java;while;cnt:40000000;lt:49
Java;forEach;cnt:40000000;lt:210
Java;ForEach + lambda;cnt:40000000;lt:157
Java;ForEach + named function;cnt:40000000;lt:1985
Java;Recursive;cnt:40000000;lt:105
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 35
Js;ForEach + lambda function;cnt: 40000000 ;lt: 240
Js;ForEach + named function;cnt: 40000000 ;lt: 233
Js;Recursive;cnt: 40000000 ;lt: 264
Pas;for;cnt:40000000;lt:    80
Pas;while;cnt:40000000;lt:    68
Pas;forEach;cnt:40000000;lt:   557
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   147
Py;For;cnt: 40000000 ;lt: 1180
Py;While;cnt: 40000000 ;lt: 1764
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 320
Py;ForEach + named function;cnt: 40000000 ;lt: 3195
Py;recursive;cnt: 40000000 ;lt: 9001

Cpp;while;cnt:40000000;lt:2.975
Cpp;do-while;cnt:40000000;lt:3.126
Cpp;for;cnt:40000000;lt:4.612
Cpp;ForEach + lambda;cnt:40000000;lt:1122.08
Cpp;ForEach + named function;cnt:40000000;lt:1115.68
Cpp;Recursive;cnt:40000000;lt:125.834
cs;for;cnt:40000000;lt:24
cs;while;cnt:40000000;lt:19
cs;ForEach + lambda;cnt:40000000;lt:91
cs;ForEach + named function;cnt:40000000;lt:89
cs;Recursive;cnt:40000000;lt:86
Java;for;cnt:40000000;lt:37
Java;while;cnt:40000000;lt:47
Java;forEach;cnt:40000000;lt:222
Java;ForEach + lambda;cnt:40000000;lt:148
Java;ForEach + named function;cnt:40000000;lt:1964
Java;Recursive;cnt:40000000;lt:102
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 33
```

```
Js;ForEach + lambda function;cnt: 40000000 ;lt: 240
Js;ForEach + named function;cnt: 40000000 ;lt: 232
Js;Recursive;cnt: 40000000 ;lt: 262
Pas;for;cnt:40000000;lt:    82
Pas;while;cnt:40000000;lt:    64
Pas;forEach;cnt:40000000;lt:   552
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   135
Py;For;cnt: 40000000 ;lt: 1146
Py;While;cnt: 40000000 ;lt: 1817
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 193
Py;ForEach + named function;cnt: 40000000 ;lt: 3313
Py;recursive;cnt: 40000000 ;lt: 11275

Cpp;while;cnt:40000000;lt:3.112
Cpp;do-while;cnt:40000000;lt:1.315
Cpp;for;cnt:40000000;lt:3.389
Cpp;ForEach + lambda;cnt:40000000;lt:1113.93
Cpp;ForEach + named function;cnt:40000000;lt:1119.23
Cpp;Recursive;cnt:40000000;lt:128.719
cs;for;cnt:40000000;lt:21
cs;while;cnt:40000000;lt:19
cs;ForEach + lambda;cnt:40000000;lt:92
cs;ForEach + named function;cnt:40000000;lt:92
cs;Recursive;cnt:40000000;lt:87
Java;for;cnt:40000000;lt:55
Java;while;cnt:40000000;lt:50
Java;forEach;cnt:40000000;lt:241
Java;ForEach + lambda;cnt:40000000;lt:159
Java;ForEach + named function;cnt:40000000;lt:1982
Java;Recursive;cnt:40000000;lt:104
Js;for;cnt: 40000000 ;lt: 37
Js;While;cnt: 40000000 ;lt: 30
Js;ForEach + lambda function;cnt: 40000000 ;lt: 244
Js;ForEach + named function;cnt: 40000000 ;lt: 231
Js;Recursive;cnt: 40000000 ;lt: 261
Pas;for;cnt:40000000;lt:    87
Pas;while;cnt:40000000;lt:    66
Pas;forEach;cnt:40000000;lt:   554
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   133
Py;For;cnt: 40000000 ;lt: 1135
Py;While;cnt: 40000000 ;lt: 1383
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 194
Py;ForEach + named function;cnt: 40000000 ;lt: 2675
Py;recursive;cnt: 40000000 ;lt: 8004

Cpp;while;cnt:40000000;lt:7.681
Cpp;do-while;cnt:40000000;lt:1.819
Cpp;for;cnt:40000000;lt:6.874
Cpp;ForEach + lambda;cnt:40000000;lt:1107.76
Cpp;ForEach + named function;cnt:40000000;lt:1107.76
Cpp;Recursive;cnt:40000000;lt:140.189
cs;for;cnt:40000000;lt:14
cs;while;cnt:40000000;lt:24
cs;ForEach + lambda;cnt:40000000;lt:245
cs;ForEach + named function;cnt:40000000;lt:549
cs;Recursive;cnt:40000000;lt:86
Java;for;cnt:40000000;lt:37
Java;while;cnt:40000000;lt:59
Java;forEach;cnt:40000000;lt:224
Java;ForEach + lambda;cnt:40000000;lt:153
```

```
Java;ForEach + named function;cnt:40000000;lt:1981
Java;Recursive;cnt:40000000;lt:109
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 33
Js;ForEach + lambda function;cnt: 40000000 ;lt: 241
Js;ForEach + named function;cnt: 40000000 ;lt: 233
Js;Recursive;cnt: 40000000 ;lt: 259
Pas;for;cnt:40000000;lt:    82
Pas;while;cnt:40000000;lt:    64
Pas;forEach;cnt:40000000;lt:   549
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   133
Py;For;cnt: 40000000 ;lt: 1131
Py;While;cnt: 40000000 ;lt: 1825
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 664
Py;ForEach + named function;cnt: 40000000 ;lt: 3704
Py;recursive;cnt: 40000000 ;lt: 10454

Cpp;while;cnt:40000000;lt:1.2
Cpp;do-while;cnt:40000000;lt:1.137
Cpp;for;cnt:40000000;lt:0.599
Cpp;ForEach + lambda;cnt:40000000;lt:1111.89
Cpp;ForEach + named function;cnt:40000000;lt:1113.95
Cpp;Recursive;cnt:40000000;lt:127.868
cs;for;cnt:40000000;lt:29
cs;while;cnt:40000000;lt:19
cs;ForEach + lambda;cnt:40000000;lt:94
cs;ForEach + named function;cnt:40000000;lt:91
cs;Recursive;cnt:40000000;lt:93
Java;for;cnt:40000000;lt:23
Java;while;cnt:40000000;lt:36
Java;forEach;cnt:40000000;lt:203
Java;ForEach + lambda;cnt:40000000;lt:158
Java;ForEach + named function;cnt:40000000;lt:1969
Java;Recursive;cnt:40000000;lt:114
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 33
Js;ForEach + lambda function;cnt: 40000000 ;lt: 243
Js;ForEach + named function;cnt: 40000000 ;lt: 230
Js;Recursive;cnt: 40000000 ;lt: 261
Pas;for;cnt:40000000;lt:    90
Pas;while;cnt:40000000;lt:    60
Pas;forEach;cnt:40000000;lt:   576
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   126
Py;For;cnt: 40000000 ;lt: 1141
Py;While;cnt: 40000000 ;lt: 1847
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 52
Py;ForEach + named function;cnt: 40000000 ;lt: 3655
Py;recursive;cnt: 40000000 ;lt: 8668

Cpp;while;cnt:40000000;lt:4.09
Cpp;do-while;cnt:40000000;lt:2.544
Cpp;for;cnt:40000000;lt:6.484
Cpp;ForEach + lambda;cnt:40000000;lt:1115.32
Cpp;ForEach + named function;cnt:40000000;lt:1108.97
Cpp;Recursive;cnt:40000000;lt:110.164
cs;for;cnt:40000000;lt:16
cs;while;cnt:40000000;lt:31
cs;ForEach + lambda;cnt:40000000;lt:113
cs;ForEach + named function;cnt:40000000;lt:81
cs;Recursive;cnt:40000000;lt:89
```

```
Java;for;cnt:40000000;lt:51
Java;while;cnt:40000000;lt:63
Java;forEach;cnt:40000000;lt:220
Java;ForEach + lambda;cnt:40000000;lt:148
Java;ForEach + named function;cnt:40000000;lt:1965
Java;Recursive;cnt:40000000;lt:115
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 35
Js;ForEach + lambda function;cnt: 40000000 ;lt: 243
Js;ForEach + named function;cnt: 40000000 ;lt: 256
Js;Recursive;cnt: 40000000 ;lt: 253
Pas;for;cnt:40000000;lt:    90
Pas;while;cnt:40000000;lt:    66
Pas;forEach;cnt:40000000;lt:   538
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   135
Py;For;cnt: 40000000 ;lt: 1190
Py;While;cnt: 40000000 ;lt: 1833
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 270
Py;ForEach + named function;cnt: 40000000 ;lt: 4329
Py;recursive;cnt: 40000000 ;lt: 9364

Cpp;while;cnt:40000000;lt:16.581
Cpp;do-while;cnt:40000000;lt:10.326
Cpp;for;cnt:40000000;lt:17.516
Cpp;ForEach + lambda;cnt:40000000;lt:1101.75
Cpp;ForEach + named function;cnt:40000000;lt:1132.63
Cpp;Recursive;cnt:40000000;lt:135.849
cs;for;cnt:40000000;lt:26
cs;while;cnt:40000000;lt:31
cs;ForEach + lambda;cnt:40000000;lt:101
cs;ForEach + named function;cnt:40000000;lt:90
cs;Recursive;cnt:40000000;lt:88
Java;for;cnt:40000000;lt:40
Java;while;cnt:40000000;lt:60
Java;forEach;cnt:40000000;lt:196
Java;ForEach + lambda;cnt:40000000;lt:201
Java;ForEach + named function;cnt:40000000;lt:1931
Java;Recursive;cnt:40000000;lt:108
Js;for;cnt: 40000000 ;lt: 35
Js;While;cnt: 40000000 ;lt: 32
Js;ForEach + lambda function;cnt: 40000000 ;lt: 251
Js;ForEach + named function;cnt: 40000000 ;lt: 239
Js;Recursive;cnt: 40000000 ;lt: 260
Pas;for;cnt:40000000;lt:    76
Pas;while;cnt:40000000;lt:    65
Pas;forEach;cnt:40000000;lt:   561
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   126
Py;For;cnt: 40000000 ;lt: 2125
Py;While;cnt: 40000000 ;lt: 1330
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 2215
Py;ForEach + named function;cnt: 40000000 ;lt: 4767
Py;recursive;cnt: 40000000 ;lt: 9615

Cpp;while;cnt:40000000;lt:9.26
Cpp;do-while;cnt:40000000;lt:7.192
Cpp;for;cnt:40000000;lt:2.262
Cpp;ForEach + lambda;cnt:40000000;lt:1122.75
Cpp;ForEach + named function;cnt:40000000;lt:1137.29
Cpp;Recursive;cnt:40000000;lt:123.278
cs;for;cnt:40000000;lt:18
```

```
cs;while;cnt:40000000;lt:16
cs;ForEach + lambda;cnt:40000000;lt:113
cs;ForEach + named function;cnt:40000000;lt:111
cs;Recursive;cnt:40000000;lt:78
Java;for;cnt:40000000;lt:54
Java;while;cnt:40000000;lt:48
Java;forEach;cnt:40000000;lt:230
Java;ForEach + lambda;cnt:40000000;lt:161
Java;ForEach + named function;cnt:40000000;lt:1995
Java;Recursive;cnt:40000000;lt:103
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 35
Js;ForEach + lambda function;cnt: 40000000 ;lt: 245
Js;ForEach + named function;cnt: 40000000 ;lt: 235
Js;Recursive;cnt: 40000000 ;lt: 256
Pas;for;cnt:40000000;lt:   102
Pas;while;cnt:40000000;lt:    53
Pas;forEach;cnt:40000000;lt:   564
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   134
Py;For;cnt: 40000000 ;lt: 1201
Py;While;cnt: 40000000 ;lt: 2038
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 578
Py;ForEach + named function;cnt: 40000000 ;lt: 4587
Py;recursive;cnt: 40000000 ;lt: 8701

Cpp;while;cnt:40000000;lt:12.72
Cpp;do-while;cnt:40000000;lt:9.18
Cpp;for;cnt:40000000;lt:1.849
Cpp;ForEach + lambda;cnt:40000000;lt:1101.42
Cpp;ForEach + named function;cnt:40000000;lt:1105.52
Cpp;Recursive;cnt:40000000;lt:124.376
cs;for;cnt:40000000;lt:4
cs;while;cnt:40000000;lt:10
cs;ForEach + lambda;cnt:40000000;lt:124
cs;ForEach + named function;cnt:40000000;lt:105
cs;Recursive;cnt:40000000;lt:92
Java;for;cnt:40000000;lt:53
Java;while;cnt:40000000;lt:51
Java;forEach;cnt:40000000;lt:220
Java;ForEach + lambda;cnt:40000000;lt:167
Java;ForEach + named function;cnt:40000000;lt:1988
Java;Recursive;cnt:40000000;lt:93
Js;for;cnt: 40000000 ;lt: 37
Js;While;cnt: 40000000 ;lt: 32
Js;ForEach + lambda function;cnt: 40000000 ;lt: 244
Js;ForEach + named function;cnt: 40000000 ;lt: 230
Js;Recursive;cnt: 40000000 ;lt: 259
Pas;for;cnt:40000000;lt:    87
Pas;while;cnt:40000000;lt:    53
Pas;forEach;cnt:40000000;lt:   537
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   138
Py;For;cnt: 40000000 ;lt: 1129
Py;While;cnt: 40000000 ;lt: 2036
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 246
Py;ForEach + named function;cnt: 40000000 ;lt: 2955
Py;recursive;cnt: 40000000 ;lt: 9146

Cpp;while;cnt:40000000;lt:2.277
Cpp;do-while;cnt:40000000;lt:11.445
Cpp;for;cnt:40000000;lt:14.63
```

```
Cpp;ForEach + lambda;cnt:40000000;lt:1127.67
Cpp;ForEach + named function;cnt:40000000;lt:1130.66
Cpp;Recursive;cnt:40000000;lt:118.777
cs;for;cnt:40000000;lt:17
cs;while;cnt:40000000;lt:14
cs;ForEach + lambda;cnt:40000000;lt:93
cs;ForEach + named function;cnt:40000000;lt:89
cs;Recursive;cnt:40000000;lt:95
Java;for;cnt:40000000;lt:45
Java;while;cnt:40000000;lt:58
Java;forEach;cnt:40000000;lt:216
Java;ForEach + lambda;cnt:40000000;lt:175
Java;ForEach + named function;cnt:40000000;lt:1961
Java;Recursive;cnt:40000000;lt:99
Js;for;cnt: 40000000 ;lt: 37
Js;While;cnt: 40000000 ;lt: 32
Js;ForEach + lambda function;cnt: 40000000 ;lt: 248
Js;ForEach + named function;cnt: 40000000 ;lt: 259
Js;Recursive;cnt: 40000000 ;lt: 258
Pas;for;cnt:40000000;lt:    79
Pas;while;cnt:40000000;lt:    68
Pas;forEach;cnt:40000000;lt:   528
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   135
Py;For;cnt: 40000000 ;lt: 3913
Py;While;cnt: 40000000 ;lt: 1955
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 262
Py;ForEach + named function;cnt: 40000000 ;lt: 3428
Py;recursive;cnt: 40000000 ;lt: 9675

Cpp;while;cnt:40000000;lt:6.028
Cpp;do-while;cnt:40000000;lt:1.992
Cpp;for;cnt:40000000;lt:3.412
Cpp;ForEach + lambda;cnt:40000000;lt:1157.15
Cpp;ForEach + named function;cnt:40000000;lt:1125.75
Cpp;Recursive;cnt:40000000;lt:130.32
cs;for;cnt:40000000;lt:21
cs;while;cnt:40000000;lt:16
cs;ForEach + lambda;cnt:40000000;lt:99
cs;ForEach + named function;cnt:40000000;lt:92
cs;Recursive;cnt:40000000;lt:90
Java;for;cnt:40000000;lt:53
Java;while;cnt:40000000;lt:50
Java;forEach;cnt:40000000;lt:218
Java;ForEach + lambda;cnt:40000000;lt:164
Java;ForEach + named function;cnt:40000000;lt:1934
Java;Recursive;cnt:40000000;lt:111
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 35
Js;ForEach + lambda function;cnt: 40000000 ;lt: 258
Js;ForEach + named function;cnt: 40000000 ;lt: 243
Js;Recursive;cnt: 40000000 ;lt: 256
Pas;for;cnt:40000000;lt:    78
Pas;while;cnt:40000000;lt:    65
Pas;forEach;cnt:40000000;lt:   559
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   137
Py;For;cnt: 40000000 ;lt: 1207
Py;While;cnt: 40000000 ;lt: 1839
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 258
Py;ForEach + named function;cnt: 40000000 ;lt: 3196
Py;recursive;cnt: 40000000 ;lt: 9268
```

```
Cpp;while;cnt:40000000;lt:0.358
Cpp;do-while;cnt:40000000;lt:2.786
Cpp;for;cnt:40000000;lt:3.322
Cpp;ForEach + lambda;cnt:40000000;lt:1107.68
Cpp;ForEach + named function;cnt:40000000;lt:1109.93
Cpp;Recursive;cnt:40000000;lt:149.029
cs;for;cnt:40000000;lt:23
cs;while;cnt:40000000;lt:19
cs;ForEach + lambda;cnt:40000000;lt:96
cs;ForEach + named function;cnt:40000000;lt:92
cs;Recursive;cnt:40000000;lt:87
Java;for;cnt:40000000;lt:37
Java;while;cnt:40000000;lt:51
Java;forEach;cnt:40000000;lt:218
Java;ForEach + lambda;cnt:40000000;lt:160
Java;ForEach + named function;cnt:40000000;lt:1983
Java;Recursive;cnt:40000000;lt:96
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 33
Js;ForEach + lambda function;cnt: 40000000 ;lt: 243
Js;ForEach + named function;cnt: 40000000 ;lt: 239
Js;Recursive;cnt: 40000000 ;lt: 259
Pas;for;cnt:40000000;lt:    82
Pas;while;cnt:40000000;lt:    68
Pas;forEach;cnt:40000000;lt:   556
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   133
Py;For;cnt: 40000000 ;lt: 1177
Py;While;cnt: 40000000 ;lt: 1822
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 180
Py;ForEach + named function;cnt: 40000000 ;lt: 3663
Py;recursive;cnt: 40000000 ;lt: 9222

Cpp;while;cnt:40000000;lt:5.169
Cpp;do-while;cnt:40000000;lt:3.163
Cpp;for;cnt:40000000;lt:0.998
Cpp;ForEach + lambda;cnt:40000000;lt:1135.57
Cpp;ForEach + named function;cnt:40000000;lt:1123.54
Cpp;Recursive;cnt:40000000;lt:129.678
cs;for;cnt:40000000;lt:21
cs;while;cnt:40000000;lt:21
cs;ForEach + lambda;cnt:40000000;lt:79
cs;ForEach + named function;cnt:40000000;lt:137
cs;Recursive;cnt:40000000;lt:90
Java;for;cnt:40000000;lt:54
Java;while;cnt:40000000;lt:54
Java;forEach;cnt:40000000;lt:235
Java;ForEach + lambda;cnt:40000000;lt:161
Java;ForEach + named function;cnt:40000000;lt:1965
Java;Recursive;cnt:40000000;lt:157
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 33
Js;ForEach + lambda function;cnt: 40000000 ;lt: 243
Js;ForEach + named function;cnt: 40000000 ;lt: 237
Js;Recursive;cnt: 40000000 ;lt: 255
Pas;for;cnt:40000000;lt:    82
Pas;while;cnt:40000000;lt:    60
Pas;forEach;cnt:40000000;lt:   536
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   135
Py;For;cnt: 40000000 ;lt: 100
```

```
Py;While;cnt: 40000000 ;lt: 1208
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 498
Py;ForEach + named function;cnt: 40000000 ;lt: 3102
Py;recursive;cnt: 40000000 ;lt: 9127

Cpp;while;cnt:40000000;lt:3.758
Cpp;do-while;cnt:40000000;lt:2.042
Cpp;for;cnt:40000000;lt:6.67
Cpp;ForEach + lambda;cnt:40000000;lt:1093.56
Cpp;ForEach + named function;cnt:40000000;lt:889.726
Cpp;Recursive;cnt:40000000;lt:236.729
cs;for;cnt:40000000;lt:12
cs;while;cnt:40000000;lt:24
cs;ForEach + lambda;cnt:40000000;lt:264
cs;ForEach + named function;cnt:40000000;lt:141
cs;Recursive;cnt:40000000;lt:73
Java;for;cnt:40000000;lt:53
Java;while;cnt:40000000;lt:51
Java;forEach;cnt:40000000;lt:313
Java;ForEach + lambda;cnt:40000000;lt:197
Java;ForEach + named function;cnt:40000000;lt:2064
Java;Recursive;cnt:40000000;lt:106
Js;for;cnt: 40000000 ;lt: 38
Js;While;cnt: 40000000 ;lt: 32
Js;ForEach + lambda function;cnt: 40000000 ;lt: 246
Js;ForEach + named function;cnt: 40000000 ;lt: 239
Js;Recursive;cnt: 40000000 ;lt: 264
Pas;for;cnt:40000000;lt:    90
Pas;while;cnt:40000000;lt:    67
Pas;forEach;cnt:40000000;lt:   558
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   135
Py;For;cnt: 40000000 ;lt: 1199
Py;While;cnt: 40000000 ;lt: 1826
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 285
Py;ForEach + named function;cnt: 40000000 ;lt: 3359
Py;recursive;cnt: 40000000 ;lt: 11310

Cpp;while;cnt:40000000;lt:4.049
Cpp;do-while;cnt:40000000;lt:1.115
Cpp;for;cnt:40000000;lt:20.573
Cpp;ForEach + lambda;cnt:40000000;lt:1223.32
Cpp;ForEach + named function;cnt:40000000;lt:1120.09
Cpp;Recursive;cnt:40000000;lt:128.421
cs;for;cnt:40000000;lt:9
cs;while;cnt:40000000;lt:18
cs;ForEach + lambda;cnt:40000000;lt:130
cs;ForEach + named function;cnt:40000000;lt:99
cs;Recursive;cnt:40000000;lt:88
Java;for;cnt:40000000;lt:54
Java;while;cnt:40000000;lt:51
Java;forEach;cnt:40000000;lt:216
Java;ForEach + lambda;cnt:40000000;lt:190
Java;ForEach + named function;cnt:40000000;lt:1966
Java;Recursive;cnt:40000000;lt:106
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 32
Js;ForEach + lambda function;cnt: 40000000 ;lt: 239
Js;ForEach + named function;cnt: 40000000 ;lt: 236
Js;Recursive;cnt: 40000000 ;lt: 260
Pas;for;cnt:40000000;lt:    83
Pas;while;cnt:40000000;lt:    69
Pas;forEach;cnt:40000000;lt:   553
```

```
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   141
Py;For;cnt: 40000000 ;lt: 1048
Py;While;cnt: 40000000 ;lt: 791
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 248
Py;ForEach + named function;cnt: 40000000 ;lt: 3314
Py;recursive;cnt: 40000000 ;lt: 10663

Cpp;while;cnt:40000000;lt:7.188
Cpp;do-while;cnt:40000000;lt:0.537
Cpp;for;cnt:40000000;lt:12.518
Cpp;ForEach + lambda;cnt:40000000;lt:1152.84
Cpp;ForEach + named function;cnt:40000000;lt:1147.2
Cpp;Recursive;cnt:40000000;lt:127.19
cs;for;cnt:40000000;lt:28
cs;while;cnt:40000000;lt:24
cs;ForEach + lambda;cnt:40000000;lt:107
cs;ForEach + named function;cnt:40000000;lt:91
cs;Recursive;cnt:40000000;lt:86
Java;for;cnt:40000000;lt:57
Java;while;cnt:40000000;lt:42
Java;forEach;cnt:40000000;lt:218
Java;ForEach + lambda;cnt:40000000;lt:185
Java;ForEach + named function;cnt:40000000;lt:1984
Java;Recursive;cnt:40000000;lt:111
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 34
Js;ForEach + lambda function;cnt: 40000000 ;lt: 250
Js;ForEach + named function;cnt: 40000000 ;lt: 241
Js;Recursive;cnt: 40000000 ;lt: 281
Pas;for;cnt:40000000;lt:    78
Pas;while;cnt:40000000;lt:    60
Pas;forEach;cnt:40000000;lt:   550
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   120
Py;For;cnt: 40000000 ;lt: 1062
Py;While;cnt: 40000000 ;lt: 2101
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 293
Py;ForEach + named function;cnt: 40000000 ;lt: 3142
Py;recursive;cnt: 40000000 ;lt: 9430

Cpp;while;cnt:40000000;lt:9.256
Cpp;do-while;cnt:40000000;lt:1.018
Cpp;for;cnt:40000000;lt:1.817
Cpp;ForEach + lambda;cnt:40000000;lt:1123.14
Cpp;ForEach + named function;cnt:40000000;lt:1113.97
Cpp;Recursive;cnt:40000000;lt:129.866
cs;for;cnt:40000000;lt:21
cs;while;cnt:40000000;lt:22
cs;ForEach + lambda;cnt:40000000;lt:104
cs;ForEach + named function;cnt:40000000;lt:92
cs;Recursive;cnt:40000000;lt:100
Java;for;cnt:40000000;lt:60
Java;while;cnt:40000000;lt:53
Java;forEach;cnt:40000000;lt:234
Java;ForEach + lambda;cnt:40000000;lt:162
Java;ForEach + named function;cnt:40000000;lt:1955
Java;Recursive;cnt:40000000;lt:111
Js;for;cnt: 40000000 ;lt: 36
Js;While;cnt: 40000000 ;lt: 34
Js;ForEach + lambda function;cnt: 40000000 ;lt: 240
Js;ForEach + named function;cnt: 40000000 ;lt: 234
```

```
Js;Recursive;cnt: 40000000 ;lt: 259
Pas;for;cnt:40000000;lt:    78
Pas;while;cnt:40000000;lt:    64
Pas;forEach;cnt:40000000;lt:   564
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   135
Py;For;cnt: 40000000 ;lt: 1153
Py;While;cnt: 40000000 ;lt: 1945
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 253
Py;ForEach + named function;cnt: 40000000 ;lt: 3457
Py;recursive;cnt: 40000000 ;lt: 8972

Cpp;while;cnt:40000000;lt:6.794
Cpp;do-while;cnt:40000000;lt:0.863
Cpp;for;cnt:40000000;lt:2.873
Cpp;ForEach + lambda;cnt:40000000;lt:1238.21
Cpp;ForEach + named function;cnt:40000000;lt:1205.92
Cpp;Recursive;cnt:40000000;lt:159.03
cs;for;cnt:40000000;lt:27
cs;while;cnt:40000000;lt:22
cs;ForEach + lambda;cnt:40000000;lt:167
cs;ForEach + named function;cnt:40000000;lt:180
cs;Recursive;cnt:40000000;lt:98
Java;for;cnt:40000000;lt:58
Java;while;cnt:40000000;lt:58
Java;forEach;cnt:40000000;lt:272
Java;ForEach + lambda;cnt:40000000;lt:201
Java;ForEach + named function;cnt:40000000;lt:2075
Java;Recursive;cnt:40000000;lt:110
Js;for;cnt: 40000000 ;lt: 41
Js;While;cnt: 40000000 ;lt: 35
Js;ForEach + lambda function;cnt: 40000000 ;lt: 290
Js;ForEach + named function;cnt: 40000000 ;lt: 246
Js;Recursive;cnt: 40000000 ;lt: 301
Pas;for;cnt:40000000;lt:    82
Pas;while;cnt:40000000;lt:    64
Pas;forEach;cnt:40000000;lt:   631
Pas;ForEach + lambda;cnt:40000000;lt:Not supported
Pas;ForEach + named function;cnt:40000000;lt:Not supported
Pas;Recursive;cnt:40000000;lt:   156
Py;For;cnt: 40000000 ;lt: 1375
Py;While;cnt: 40000000 ;lt: 2309
Py;ForEach [+ lambda];cnt: 40000000 ;lt: 440
Py;ForEach + named function;cnt: 40000000 ;lt: 4063
Py;recursive;cnt: 40000000 ;lt: 13387
```

Authors

MENYHÁRT László

Eötvös Loránd University, Faculty of Informatics, Department of Media and Educational Informatics, Budapest, Hungary, e-mail: menyhart@inf.elte.hu

About this document

Published in:

CENTRAL-EUROPEAN JOURNAL OF NEW TECHNOLOGIES IN RESEARCH, EDUCATION AND PRACTICE

Volume 2, Number 1. 2020.

ISSN: 2676-9425 (online)

DOI:

10.36427/CEJNTREP.2.1.421

License