

# Az ismétlés áttekintése

Menyhárt László Gábor

menyhart@inf.elte.hu  
ELTE IK

**Absztrakt.** Az ismétlés hétköznapi fogalma és az informatika általi használat, illetve a különböző implementációk eltérnek. Ebben a cikkben összeszedem a különböző szintek, paradigmák és nyelvi lehetőségeket, majd ismertetem az elvégzett mérésemet, végül didaktikai szempontok szerint is értékelem az eredményeket. Így hasznos programozó versenyre készülő diákok, illetve az őket felkészítő informatika tanárok számára.

**Kulcsszavak:** ismétlés, rekurzió, ciklus, programozási nyelvek, mérés

## 1. Bevezetés

Hétköznapi értelemben az ismétlés természetes jelenség: Gondoljunk itt a napok, hetek, évszakok stb. rendszeres váltakozására. Ugyanakkor az informatikában többféleképpen jelenik meg. Máshogy gondolkozhatunk, máshogy fordul elő a különböző paradigmákban, a különböző nyelveken más a szintaxis és máshogy lett implementálva, ezért a futásuk is különböző, vagyis a futásidejük is nagyban eltér. Ezen észrevételek és benyomások ösztönöztek arra, hogy gyűjtsem össze a jelenlegi lehetőségeket és hasonlítsam össze őket.

Ebben a cikkben a fogalmi meghatározásoktól kezdve, különböző implementációk futtatásának mérésén keresztül a didaktikai megfontolásokig áttekintem az eredményeket. A bemutatásra kerülő módszer és a jelenlegi eredmények hasznosak lehetnek programozó versenyre készülő diákok, illetve az őket felkészítő informatika tanárok számára.

## 2. Az ismétlés fogalma

Az ismétlés valaminek a többszöri elvégzése, végrehajtása. Összegyűjtöttem pár meghatározást is különböző helyekről, annak ellenére, hogy mindenki tudja, érzi, hogy mit jelent.

A Magyar Szinonima kézikönyvtár-ban az ismét és az ismétél szavak találhatóak meg:

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

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

Az Új Magyar Lexikon-ban az ismétlés meghatározása az, hogy az „**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.]

A Magyar néprajzi lexikon [3] csak nyelvi meghatározást, mondanivaló nyomatékosítására, mint költői kifejezés ír a többszörözésről.

[4]-ben az „**ismétlés** (fn.) ... *Cselekvés, melynél fogva valamit ismét, azaz újra, még egyszer teszünk. ...*”

A Sulinet Tudásbázisában az Informatika tantárgynál az ismétlés fogalma „Többszöri végrehajtás”-ként van definiálva. [5]

A WikiSzótárban [6] az ismétlés egyik definíciója „**2. Egy cselekvés újra végzése**, amikor ugyanazt tesszük, amit korábban már megtettünk.”

A Cambridge szótárban a **repeat** szóra az szerepel, hogy „*to happen, or to do something, more than once*” [7], míg az **iteration**-re az, hogy „*the process of doing something again and again, ...*” [8].

### 3. Az ismétlés az informatika szintjein

Amikor egy konkrét megoldandó problémánk van, még csak azt tudhatjuk, hogy valamit többször kell majd végrehajtani. Gondolkozásunk következő absztrakt szintjén, már azt döntjük el, hogy egy műveletet, amit többször szeretnénk elvégezni azt kiszervezzük, majd meghívjuk önmagát, ez a rekurzió [10], vagy az adott műveletet ciklikus hívással ismétljük. Ez a megkülönböztetés már paradigmaként is megjelenik, hiszen a funkcionális programozás csak az előbbire ad lehetőséget.

A cikluson belül megkülönböztetünk feltételest és adott elemszámú lefutót.

Az adott elemszámú lefutónál két nagy csoportot különböztethetünk meg. Az egyikben egy ciklusváltozó, mint számláló értéke változik az ismétlés során, míg a másikban egy konkrét elem értékeit. Az előbbiben tehát egy sorozat elemein úgy szaladhatunk végig, hogy egy számláló növekszik egyesével és azzal indexeljük a sorozat elemeit a feldolgozáshoz, míg a másodikban rögtön a sorozat egyes elemeit kapjuk és dolgozzuk fel.

A feltételes ciklusokat is további két nagy csoportra oszthatjuk, az előtesztelő és a hátulatesztelő verzióra. Ezek között a különbség, hogy a hátulatesztelő ciklus műveletei egyszer biztosan lefutnak, míg az előtesztelőnél elképzelhető, hogy a műveletei egyszer sem hajódnak végre. Mindkettő feltételeként egy logikai kifejezést adunk meg, amittől függ, hogy az műveleteket ismét végre kell-e hajtani vagy sem. Megkülönböztetjük a ciklusokat aszerint is, hogy az igaz logikai kifejezésnél mi történik. Azaz, hogy ilyenkor a műveletek ismét lefutnak vagy pedig pont ilyenkor áll le a ciklus futása. Tehát, hogy a bent maradás vagy a kilépés feltételét kell-e megadni.

Ismétlés						
Rekurzió	Ciklus					
	Feltételes				Adott elemszámú futó	
	Előtesztelő		Hátulatesztelő		Számláló	Sorozat elemein iteráló
	bent marad	kilép	bent marad	kilép		

1. táblázat: Ismétlés áttekintő táblázata

A szintaxis bezavarhat ebbe a tiszta képbe, ugyanis a legtöbb programozási nyelv, a fenti lehetőségek közül valamelyiket nem támogatja. Például a Pascalban a kilépés feltételét kell megadni a hátulatesztelő ciklusnak, míg a C típusú nyelveknél a bent maradás feltételét. Az előtesztelő ciklusnál általában a bent maradás feltételét kell megadni, de a Linux-os shell scriptben van szintaktika a kilépés feltételének megadásához is. Sőt, shell scriptben mind a négyféle feltételes ciklust le tudjuk írni. C-ben a `for` kulcsszót lehet használni előtesztelő ciklusként is és ott lehetőség van a `break` illetve `continue` kulcsszavakkal az elvileg adott elemszámú futó `for` ciklus idő előtti befejezésére is. Persze a több egymásba ágyazott `for` ciklus esetén már trükközni kell, ezért érdemes elkerülni ezt a lehetőséget. Van olyan programozási nyelv is (például a Logo), ahol a ciklusváltozó meg sem jelenik, csak ha szükség van rá, akkor lehet használni (a REPEAT/ismétlés-ben a REPCOUNT/hányadik).

A magasszintű programozási nyelvekben nem csak a kulcsszavak, azaz a szintaxis térhet el, hanem különbözőféleképpen implementálják alacsony szintre, vagyis a ténylegesen futó gépi kódra a fenti lehetőségeket. Így természetesen a futásidejük is eltér. Mivel több nyelven is implementáltam [11, 12,

13, 14, 15, 16] ugyanazon feladatok megoldását és érzésre is különböző volt a futásidejük, így felmerült bennem, hogy végezzek el egy pontosabb mérést.

## 4. A mérés ismertetése

Hat különböző programozási nyelven az ismétlés öt-hat féle implementációját készítettem el. Majd a futási időket milliszekundumban mérve összegyűjtöttem.

A nyelvek a következők voltak: C++, C#, Java, JavaScript, Pascal és Python.

Implementáltam a rekurziót, elől tesztelő feltételes ciklust, számlálós ciklusok közül pedig a ciklusváltozós for-on kívül olyan iterációs változatokat, amiket az adott nyelv támogat. Párszor előfordult, hogy a nyelv nem támogatja az adott módszert, így ott az kihagyásra került. Az elemeken való végig iterálás is többféleképpen fordul elő, így ezeket egy közös csoportba vontam össze.

### 4.1. A környezet bemutatása

A mérést egy Intel® Core™ i7-8750H 2.20GHz-es processzorú, 32GB memóriával rendelkező számítógépen futtattam, melyen 64 bites Windows 10 Pro operációs rendszer futott.

A C++ fordítója a Code::Blocks 17.12-es alapértelmezett mingw32-g++-je volt, aminek a verziója 5.1.0.

A C#-hoz a Windows 10-en megtalálható .NET Framework fordítóját használtam, a „Microsoft (R) Visual C# Compiler version 4.8.3752.0”-t.

A Java-hoz a jelenleg legfrissebb 2019. október 6-án megjelent OpenJDK-t.

```
>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 kódokat a Node v10.16.0 segítségével futtattam.

Pascalhoz a 32 bites „Free Pascal Compiler version 3.0.4 [2017/10/06] for i386”-em volt.

A Python kódokat a „Python 2.7.15 (v2.7.15:ca079a3ea3, Apr 30 2018, 16:30:26) [MSC v.1500 64 bit (AMD64)]” értelmezte és futtatta.

### 4.2. Az implementált algoritmus bemutatása

Mind a hat programozási nyelven az összes lehetséges módon a következő algoritmus szerint implementáltam az ismétlést.

Mivel kiderült, hogy a rekurzió lépésszáma a vizsgált programozási nyelvekben korlátos, ezért először annak a maximális lehetséges lépésszámának a meghatározásával kezdtem. A 4.3.-as fejezetben lesznek a pontos értékek. Az így kimért legkisebb lépésszámot használtam az összes módszernél. Mivel így nagyon gyorsan lefutott a kód ezt többször kellett elvégezni, amihez az összes nyelven egy számlálós (for) ciklust választottam.

Az eltelt idő mérését a futás előtti és a futás utáni időpont különbségéből számoltam ki. Üres, azaz műveletet nem tartalmazó ciklust nem érdemes csinálni, ezért egy egyszerű értékadást tettem bele. Viszont engem csak a ciklusra szánt idő érdekelt, ezért az értékadásra szánt időt ki kellett találnom. Ezt pedig úgy csináltam, hogy rögtön még egyszer megismételtem ugyanazt a ciklust két értékadással is. Az absztrakt algoritmus:

```
time1
loop
  assignment1
time2
```

```

loop
  assignment1
  assignment2
time3

```

Kiszámoltam, hogy mennyi az egy értékadásra szánt teljes idő (1) és feltettem, hogy az értékadások ugyanannyi idő alatt futnak le (2). A ciklusra szánt konkrét időt (3) tehát így számoltam ki:

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

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

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

Az implementált kódok az A. mellékletben találhatóak.

### 4.3. Mérési eredmények

Először meghatároztam a különböző nyelveken történő rekurzív megvalósítás alapján egy lépésszámot, azt a legnagyobbat, amelyik mindegyiken elfut. A rekurzív függvény implementálásakor az utasítások elejére tettem a rekurzív hívást, hogy a fordító biztosan ne optimalizálja a fark-rekurziót.

Programozási nyelv	MaxCnt (Ebben a környezetben, ezen programok esetében)
C++	56988
C#	15918
Java	11420
JavaScript	8940
Pascal	65134
Python	4194

2. táblázat: Rekurzív hívások maximális száma

Végül 4000 ismétlést egy számláló ciklussal, azaz az implementációkban `for` szintaktikával 10.000-szer megismételtem, hogy mérhető és összehasonlítható eredményt kapjak. Később kiderült, hogy ez jó választás volt, mert ez az egyik leggyorsabb, így nem zavar bele annyira az összehasonlításba. Így kétszer 40.000.000 szer futott minden mért ismétlés.

### 4.4. Mérés kiértékelés

A B. mellékletben elérhető nyers adatokból Excel segítségével átlagokat és szórást számoltam. Az időt ( $T_l$ ) milliszekundumban írtam ki és az első összesen 40 millió ismétlésre vonatkozik. A szórást az átlag értékhez viszonyított százalékos értékkel adtam meg, hogy könnyebben érthető és összehasonlítható legyen. Az eredmények a következő táblázatban láthatóak:

Programozási nyelv	While	For	ForEach [+lambda]	ForEach +named fnc	Rekurzív
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%)

3. táblázat: Mérési eredmények: átlag (szórás az átlaghoz viszonyítva %-ban)

#### 4.5. Az eredmény véleményezése

A táblázatból látszik, hogy az előtesztelő ciklus majdnem mindig a leggyorsabb, de az adott lépésszámú ciklusváltozós megoldás is elhanyagolható lassulást okozott három nyelvénél, míg a Python esetében kétszer gyorsabb. Sebességben a rekurzív hívás követi az előzőeket, ami Pascalban és Javában 2-szer, C#-ban, JavaScriptben és Pythonban 4-5-ször, C++-ban pedig körülbelül 20-szor lassabb, mint az előzőek átlaga. Az eddigi idők öt programozási nyelvénél a másodperc töredéke, míg Pythonban már 10 másodperc körüli eredményt mutat. A sorozat elemein történő végig iterálás tekintetében a C# és JavaScript összemérhető a rekurzív hívás idejével, de a C++ 9-szer lassabb a rekurzív hívásnál, míg a Java 2-szer vagy 18-szor lassabb, attól függően, hogy anonymous vagy elnevezett függvényt használunk. Míg Pythonban a számlálós ciklushoz képest „csak” 3-szoros lassulást, de a rekurzív híváshoz képest gyorsulást mértem. Az ismétlés a C++ számlálós ciklusával a leggyorsabb, míg a Pythonban a rekurzív megvalósítása a leglassabb.

### 5. Didaktikai megfontolások

Ha oktatáshoz a fenti táblázat alapján kellene nyelvet választanom, akkor a C#-ot választanám, mert az egyenletesen hoz elég jó időket. A Pascal-t nem választanám, mert nem támogat mindent. A Python meg nagyon lassú. Tehát a sorrend C#, JavaScript, C++ és Java lenne az ismétlés különböző implementációinak futási ideje és azok szórása alapján.

Természetesen mindegyik lehetőséget meg kell tanítanunk hallgatóinknak és meg kell mutatnunk, hogy mi alapján tudnak választani a paradigmák, szintaxisok között.

Vannak esetek, amikor a kód olvashatósága fontosabb, de előfordul, hogy a futási sebességet kell előtérbe helyezni. Például már többen megállapították [9], hogy a rekurzív eljárások többnyire kevésbé hatékonyak, mint az iterációt használók, de gyakran egyszerűbb, könnyebben olvasható leírást kapunk. Egy ilyen összehasonlító táblázat hasznos lehet olyan programozási versenyeken is, ahol időlimitet használnak és a versenyző választhatja meg a nyelvet. Habár ott legtöbbször nem csak egy értékadás a művelet és a többi nyelvi elem is befolyásolhatja az futás összes idejét.

A cikkemben megtalálható tudás és mérési eredmények segíthetik az adott feladathoz történő megfelelő implementáció kiválasztását. A mérés technikája pedig később jól jöhet, hiszen a jövőben a processzorok fejlődése, illetve az újabb verziójú fordítók megjelenése változtathat az itt ismertetett időközön, vagyis újra el kell végezni ezt a Benchmark-ot [17]. Hiszen az újabb fordító verziók a kódok elemzésével egyre hatékonyabb bináris kódokat hozhatnak létre, akár a kódok különböző mértékű változtatásával. Például a Python-ból már most is elérhető a 3.8-as, sőt több alternatív implementáció is létezik hozzá [18, 19], amiket még érdemes lesz összehasonlítani.

## Irodalom

1. Bék Gerzson, Csiffáry 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 magyar akadémiai nyomdásznál Pest (1862)  
<https://czuczor.oszk.hu/kereses.php?kereses=ism%C3%A9tl%C3%A9s> (utoljára megtekintve: 2019.11.11.)  
[https://mek.oszk.hu/05800/05887/pdf/3kotet\\_1.pdf](https://mek.oszk.hu/05800/05887/pdf/3kotet_1.pdf) (74. oldal, utoljára megtekintve: 2019.11.11.)
5. *Ismétlés fogalma a Sulinet tudásbázisban* (2016)  
<https://tudasbazis.sulinet.hu/hu/informatika/informatika/informatika-2-evfolyam/algorithmusok-a-het-koznapiokban-tevekenysegek-elemekre-bontasa-helyes-sorrend-megkerese/ismetles-szerepe> (utoljára megtekintve: 2019.11.11.)
6. *WikiSzótár* (2012)  
<https://wikiszotar.hu/ertelmezo-szotar/Ism%C3%A9tl%C3%A9s> (utoljára megtekintve: 2019.11.11.)
7. *Repeat in Cambridge Dictionary* (2019)  
<https://dictionary.cambridge.org/dictionary/english/repeat> (utoljára megtekintve: 2019.11.11.)
8. *Iteration in Cambridge Dictionary* (2019)  
<https://dictionary.cambridge.org/dictionary/english/iteration> (utoljára megtekintve: 2019.11.11.)
9. Rónyai L., Ivanyos G., Szabó R.: *Algoritmusok*, Typotex (2004) 37. oldal
10. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest.: *Introduction to Algorithms*, MIT Press (1990) (magyar verzió: *Algoritmusok*, Műszaki Könyvkiadó 1997)
11. *C++ referencia* (2019)  
<https://en.cppreference.com/w/> (utoljára megtekintve: 2019.11.11.)
12. *C# referencia* (2017)  
<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/> (utoljára megtekintve: 2019.11.11.)
13. *Java documentation* (2019)  
<https://docs.oracle.com/javase/tutorial/java/> (utoljára megtekintve: 2019.11.11.)
14. *JavaScript reference* (2019)  
<https://developer.mozilla.org/hu/docs/Web/JavaScript/Reference> (utoljára megtekintve: 2019.11.11.)
15. *Free Pascal reference guide* (2017)  
<https://www.freepascal.org/docs-html/ref/ref.html> (utoljára megtekintve: 2019.11.11.)
16. *The Python Language Reference* (2019)  
<https://docs.python.org/2.7/reference/index.html> (utoljára megtekintve: 2019.11.11.)
17. *Benchmark jelentése*  
<https://pcforum.hu/szotar/Benchmark> (utoljára megtekintve: 2019.11.11.)
18. *Alternative Python Implementations*  
<https://www.python.org/download/alternatives/> (utoljára megtekintve: 2019.11.11.)
19. *Cython*  
<https://cython.org/> (utoljára megtekintve: 2019.11.11.)

## Mellékletek

### A. Forráskódok

#### 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-elap-
sed.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:" << elap-
sed.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 func-
tion;cnt:{1};lt:{0}", elapsed.Milliseconds-(elapsed2.Milliseconds-el-
apsed.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.Milli-
seconds-(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- (elap-
sed2-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- (el-
apsed2-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:"+ti-
mes*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:"+ti-
mes*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-elap-
sed));

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-elap-
sed));

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:",el-
apsed-(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:",elap-
sed-(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-elap-
sed))*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-elap-
sed))*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-el-
apsed))*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 sup-
ported');
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/1000)-((elapsed2.seconds*1000+elapsed2.microseconds/1000)-(elapsed.seconds*1000+elapsed.microseconds/1000))

```

## B. A mérés nyers adatai

```

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
```