

# **PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

GABRIEL VARGAS BENTO DE SOUZA  
NILSON DEON CORDEIRO FILHO  
SAULO DE MOURA ZANDONA FREITAS

## **EXERCÍCIO PRÁTICO 04** Arquitetura de Computadores II

Belo Horizonte - MG  
2023

## 1.0a Tabelas Arduíno – Gabriel Vargas

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		Soma	Or	Mult	Soma	Or	Mult
byte	2462960	2526108	2526108	2652396	2652396	2652396	2841836
int	2715548	2841836	2778688	3031276	3094424	3094424	3599592
float	3220712	12437840	XXXXXXXX	10356416	12690432	XXXXXXXX	10609008

Tempos (us) obtidos pelo arduíno

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		Soma	Or	Mult	Soma	Or	Mult
byte	2462960	63148	63148	189436	189436	189436	378876
int	2715548	126288	63140	315728	378876	378876	884044
float	3220712	9217128	XXXXXXXX	7135704	9469720	XXXXXXXX	7388296

Tempos (us) de cada operação separada obtidos (menos tempo base) pelo arduíno

Tipo	MIPS (ATM328P)					
	Constante (Ex.: i = i op 3;)			Constante (Ex.: i = i op j;)		
	Soma	Or	Mult	Soma	Or	Mult
byte	15.8358	15.8358	5.2788	5.2788	5.2788	2.6394
int	7.9184	15.8378	3.1673	2.6394	2.6394	1.1312
Tipo	MFLOPS (ATM328P)					
	Constante (Ex.: i = i op 3;)			Constante (Ex.: i = i op j;)		
	Soma	Or	Mult	Soma	Or	Mult
float	0.1085	XXXXXXXX	0.1401	0.1056	XXXXXXXX	0.1353

MIPS e MFLOPS calculados a partir dos tempos obtidos

Tipo	CPI					
	Soma	Or	Mult	Soma	Or	Mult
char	1.010368	1.010368	3.030976	3.030976	3.030976	6.062016
int	2.020608	1.01024	5.051648	6.062016	6.062016	14.144704
float	147.474048	XXXXXXXX	114.171264	151.51552	XXXXXXXX	118.212736

CPI calculado a partir dos tempos obtidos

## 1.0b Tabelas Arduíno – Nilson Deon

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		Soma	Or	Mult	Soma	Or	Mult
byte	2462940	2526092	2526092	2652380	2652156	2652380	2841596
int	2715532	2841820	2778672	3031260	3094168	3094408	3599348
float	3220696	12690416	XXXXXXXX	10356404	12690176	XXXXXXXX	10356404

Tempos (us) obtidos pelo arduíno

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		Soma	Or	Mult	Soma	Or	Mult
byte	2462940	63152	63152	189440	189216	189440	378656
int	2715532	126288	63140	315728	378636	378876	883816
float	3220696	9469720	XXXXXXXX	7135708	9469480	XXXXXXXX	7135708

Tempos (us) de cada operação separada obtidos (menos tempo base) pelo arduíno

Tipo	MIPS (ATM328P)					
	Constante (Ex.: i = i op 3;)			Constante (Ex.: i = i op j;)		
	Soma	Or	Mult	Soma	Or	Mult
byte	15.8348	15.8348	5.2787	5.2850	5.2787	2.6409
int	7.9184	15.8378	3.1673	2.6411	2.6394	1.1315
Tipo	MFLOPS (ATM328P)					
	Constante (Ex.: i = i op 3;)			Constante (Ex.: i = i op j;)		
	Soma	Or	Mult	Soma	Or	Mult
float	0.1056	XXXXXXXX	0.1401	0.1056	XXXXXXXX	0.1401

MIPS e MFLOPS calculados a partir dos tempos obtidos

Tipo	CPI					
	Soma	Or	Mult	Soma	Or	Mult
char	1.010432	1.010432	3.03104	3.027456	3.03104	6.058496
int	2.020608	1.01024	5.051648	6.058176	6.062016	14.141056
float	151.51552	XXXXXXXX	114.171328	151.51168	XXXXXXXX	114.171328

CPI calculado a partir dos tempos obtidos

### 1.0c Tabelas Arduíno – Saulo de Moura

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		Soma	Or	Mult	Soma	Or	Mult
byte	2210364	2526092	2399792	2652380	2652380	2841824	2841824
int	2210364	2841820	2589240	3031260	3094408	3220708	3599576
float	2210364	12437824	XXXXXXXX	10356400	12690416	XXXXXXXX	10608980

Tempos (us) obtidos pelo arduíno

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		Soma	Or	Mult	Soma	Or	Mult
byte	2210364	315728	189428	442016	442016	631460	631460
int	2210364	631456	378876	820896	884044	1010344	1389212
float	2210364	10227460	XXXXXXXX	8146036	10480052	XXXXXXXX	8398616

Tempos (us) de cada operação separada obtidos (menos tempo base) pelo arduíno

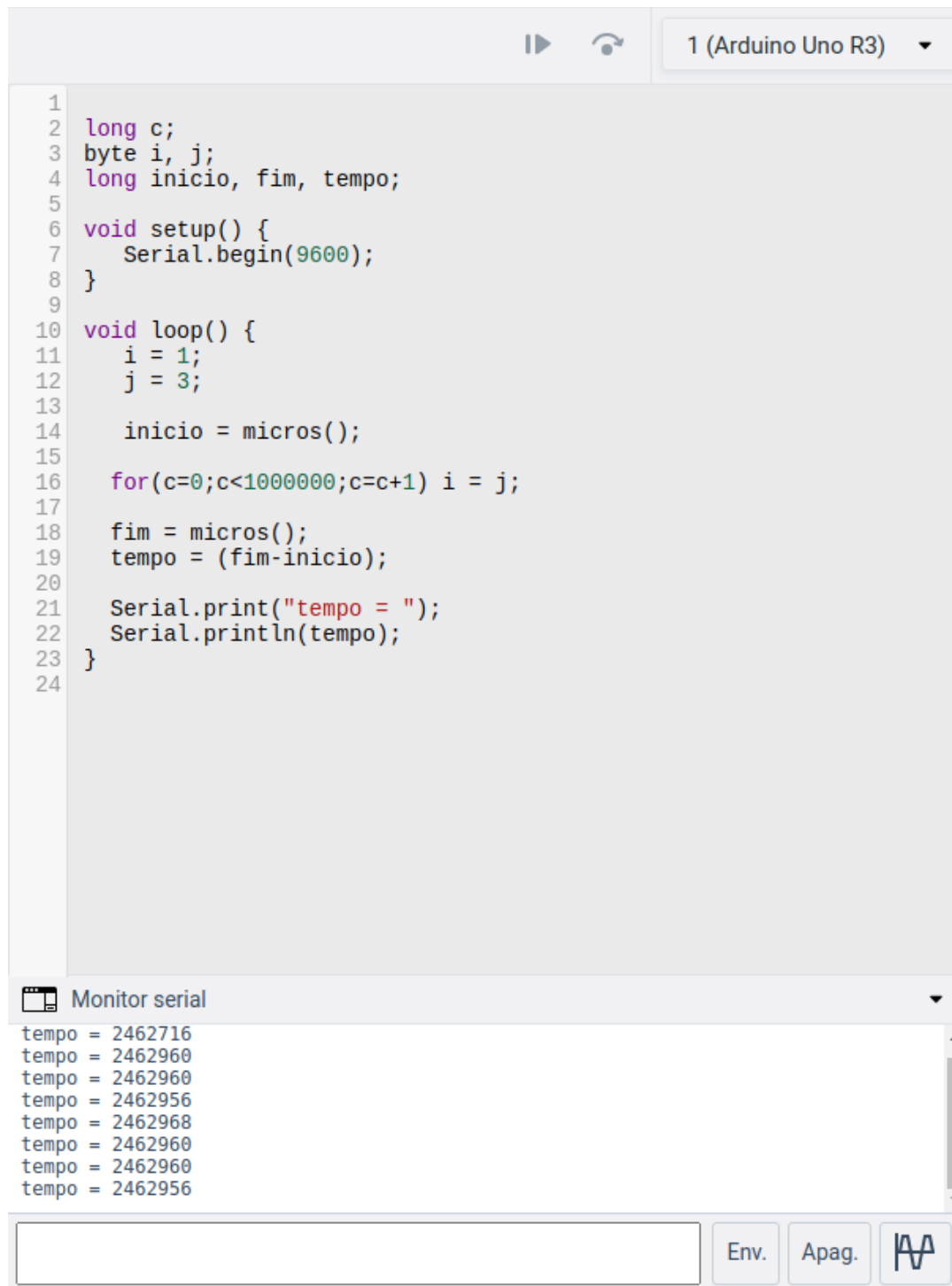
Tipo	MIPS (ATM328P)					
	Constante (Ex.: i = i op 3;)			Constante (Ex.: i = i op j;)		
	Soma	Or	Mult	Soma	Or	Mult
byte	3.1673	5.2791	2.2624	2.2624	1.5836	1.5836
int	1.5836	2.6394	1.2182	1.1312	0.9898	0.7198
Tipo	MFLOPS (ATM328P)					
	Constante (Ex.: i = i op 3;)			Constante (Ex.: i = i op j;)		
	Soma	Or	Mult	Soma	Or	Mult
float	0.0978	XXXXXXXX	0.1228	0.0954	XXXXXXXX	0.1191

MIPS e MFLOPS calculados a partir dos tempos obtidos

Tipo	CPI					
	Soma	Or	Mult	Soma	Or	Mult
char	5.051648	3.030848	7.072256	7.072256	10.10336	10.10336
int	10.103296	6.062016	13.134336	14.144704	16.165504	22.227392
float	163.63936	XXXXXXXX	130.336576	167.680832	XXXXXXXX	134.377856

CPI calculado a partir dos tempos obtidos

## 2.0 Prints testes Arduíno



The screenshot displays the Arduino IDE interface. At the top, the board is set to "1 (Arduino Uno R3)". The main editor window contains the following C++ code:

```
1
2 long c;
3 byte i, j;
4 long inicio, fim, tempo;
5
6 void setup() {
7     Serial.begin(9600);
8 }
9
10 void loop() {
11     i = 1;
12     j = 3;
13
14     inicio = micros();
15
16     for(c=0;c<1000000;c=c+1) i = j;
17
18     fim = micros();
19     tempo = (fim-inicio);
20
21     Serial.print("tempo = ");
22     Serial.println(tempo);
23 }
24
```

Below the code editor is the "Monitor serial" window, which shows the output of the program:

```
tempo = 2462716
tempo = 2462960
tempo = 2462960
tempo = 2462956
tempo = 2462968
tempo = 2462960
tempo = 2462960
tempo = 2462956
```

At the bottom of the serial monitor, there is an input field and three buttons: "Env.", "Apag.", and a button with a waveform icon.

Print teste tempo base para byte

1 (Arduino Uno R3) ▾

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

```
long c;  
int i, j;  
long inicio, fim, tempo;  
  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  i = 1;  
  j = 3;  
  
  inicio = micros();  
  
  for(c=0;c<1000000;c=c+1) i = i + 3;  
  
  fim = micros();  
  tempo = (fim-inicio);  
  
  Serial.print("tempo = ");  
  Serial.println(tempo);  
}
```

Monitor serial ▾

```
tempo = 2841592  
tempo = 2841836  
tempo = 2841836  
tempo = 2841836  
tempo = 2841848  
tempo = 2841836  
tempo = 2841836  
tempo = 2841836
```

Env. Apag. 

Print teste tempo  $i = i + 3$  para int


1 (Arduino Uno R3) ▾

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

```
long c;  
byte i, j;  
long inicio, fim, tempo;  
  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  i = 1;  
  j = 3;  
  
  inicio = micros();  
  
  for(c=0;c<1000000;c=c+1) i = i | 3;  
  
  fim = micros();  
  tempo = (fim-inicio);  
  
  Serial.print("tempo = ");  
  Serial.println(tempo);  
}
```

Monitor serial ▾

tempo = 2525856  
tempo = 2526108  
tempo = 2526108  
tempo = 2526108  
tempo = 2526108  
tempo = 2526108  
tempo = 2526108

Env. Apag. 


Print teste tempo i = i | 3 para byte

1 (Arduino Uno R3) ▾

1  
2 long c;  
3 float i, j;  
4 long inicio, fim, tempo;  
5  
6 void setup() {  
7     Serial.begin(9600);  
8 }  
9  
10 void loop() {  
11     i = 1;  
12     j = 3;  
13  
14     inicio = micros();  
15  
16     for(c=0;c<1000000;c=c+1) i = i \* j;  
17  
18     fim = micros();  
19     tempo = (fim-inicio);  
20  
21     Serial.print("tempo = ");  
22     Serial.println(tempo);  
23 }  
24

Monitor serial ▾

tempo = 10608752  
tempo = 10609000  
tempo = 10609004  
tempo = 10608996  
tempo = 10609008  
tempo = 10608996  
tempo = 10609008

Env. Apag. 

Print teste tempo  $i = i * j$  para float



### 3.0a Tabelas Programa em C – Gabriel Vargas

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		Soma	Or	Mult	Soma	Or	Mult
char	4.5	15.4	15.1	17.0	14.7	14.8	21.2
int	2.5	15.4	15.6	17.1	14.8	14.7	22.9
float	5.4	24.0	XXXXXXXX	24.8	27.6	XXXXXXXX	25.3

Tempos (ms) de cada operação obtidos pelo programa em C (gcc 11.3.0)

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		Soma	Or	Mult	Soma	Or	Mult
char	4.5	10.9	10.6	12.5	10.2	10.3	16.7
int	2.5	12.9	13.1	14.6	12.3	12.2	20.4
float	5.4	18.6	XXXXXXXX	19.4	22.2	XXXXXXXX	19.9

Tempos (ms) de cada operação separada obtidos (menos tempo base) pelo programa em C (gcc 11.3.0)

Tipo	MIPS (Intel Core i7-10700F)					
	Constante (Ex.: i = i op 3;)			Constante (Ex.: i = i op j;)		
	Soma	Or	Mult	Soma	Or	Mult
char	917.4312	943.3962	800.0000	980.3922	970.8738	598.8024
int	775.1938	763.3588	684.9315	813.0081	819.6721	490.1961
Tipo	MFLOPS (Intel Core i7-10700F)					
	Constante (Ex.: i = i op 3;)			Constante (Ex.: i = i op j;)		
	Soma	Or	Mult	Soma	Or	Mult
float	537.6344	XXXXXXXX	515.4639	450.4505	XXXXXXXX	502.5126

MIPS e MFLOPS calculados a partir dos tempos obtidos

Tipo	CPI					
	Soma	Or	Mult	Soma	Or	Mult
byte	3.161	3.074	3.625	2.958	2.987	4.843
int	3.741	3.799	4.234	3.567	3.538	5.916
float	5.394	XXXXXXXX	5.626	6.438	XXXXXXXX	5.771

CPI calculado a partir dos tempos obtidos

### 3.0b Tabelas Programa em C – Nilson Deon

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		Soma	Or	Mult	Soma	Or	Mult
char	3.1	6.0	19.7	8.8	6.3	19.7	9.6
int	4.0	19.2	21.8	9.8	19.7	20.6	8.8
float	5.7	23.6	XXXXXXXX	1005.7	24.3	XXXXXXXX	1305.6

Tempos (ms) de cada operação obtidos pelo programa em C (gcc 11.3.0)

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		Soma	Or	Mult	Soma	Or	Mult
char	3.1	2.9	16.6	5.7	3.2	16.6	6.5
int	4.0	15.2	17.8	5.8	15.7	16.6	4.8
float	5.7	17.9	XXXXXXXX	1000	18.6	XXXXXXXX	1299.9

Tempos (ms) de cada operação separada obtidos (menos tempo base) pelo programa em C (gcc 11.3.0)

Tipo	MIPS (Intel Core i7-10700F)					
	Constante (Ex.: i = i op 3;)			Constante (Ex.: i = i op j;)		
	Soma	Or	Mult	Soma	Or	Mult
char	3448.2759	602.4096	1754.3860	3125.0000	602.4096	1538.4615
int	657.8947	561.7978	1724.1379	636.9427	602.4096	2083.3333
Tipo	MFLOPS (Intel Core i7-10700F)					
	Constante (Ex.: i = i op 3;)			Constante (Ex.: i = i op j;)		
	Soma	Or	Mult	Soma	Or	Mult
float	558.6592	XXXXXXXX	10.0000	537.6344	XXXXXXXX	7.6929

MIPS e MFLOPS calculados a partir dos tempos obtidos

Tipo	CPI					
	Soma	Or	Mult	Soma	Or	Mult
byte	2.6378	2.5652	3.025	2.4684	2.4926	4.0414
int	3.1218	3.1702	3.5332	2.9766	2.9524	4.9368
float	4.5012	XXXXXXXX	4.6948	5.3724	XXXXXXXX	4.8158

CPI calculado a partir dos tempos obtidos

### 3.0c Tabelas Programa em C – Saulo de Moura

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		Soma	Or	Mult	Soma	Or	Mult
char	3.8	20.0	18.2	22.0	20.1	17.8	25.0
int	3.3	19.0	18.1	22.4	19.0	18.0	25.1
float	5.5	27.2	XXXXXXXX	27.5	28	XXXXXXXX	27.2

**Figura 09c:** Tempos (ms) de cada operação obtidos pelo programa em C (gcc 11.3.0)

Tipo	Tempo base	Use para o teste (i = i op 3)			Use para o teste (i = i op j)		
		Soma	Or	Mult	Soma	Or	Mult
char	3.8	16.2	14.4	18.2	16.3	14.0	21.2
int	3.3	15.7	14.8	19.1	15.7	14.7	21.8
float	5.5	21.7	XXXXXXXX	22.0	22.5	XXXXXXXX	21.7

**Figura 10c:** Tempos (ms) de cada operação separada obtidos (menos tempo base) pelo programa em C (gcc 11.3.0)

Tipo	MIPS (AMD Ryzen 5 5500U)					
	Constante (Ex.: i = i op 3;)			Constante (Ex.: i = i op j;)		
	Soma	Or	Mult	Soma	Or	Mult
char	617.2840	694.4444	549.4505	613.4969	714.2857	471.6981
int	636.9427	675.6757	523.5602	636.9427	680.2721	458.7156
Tipo	MFLOPS (AMD Ryzen 5 5500U)					
	Constante (Ex.: i = i op 3;)			Constante (Ex.: i = i op j;)		
	Soma	Or	Mult	Soma	Or	Mult
float	460.8295	XXXXXXXX	454.5455	444.4444	XXXXXXXX	460.8295

**Figura 11c:** MIPS e MFLOPS calculados a partir dos tempos obtidos

Tipo	CPI					
	Soma	Or	Mult	Soma	Or	Mult
char	3.402	3.024	3.822	3.423	2.94	4.452
int	3.297	3.108	4.011	3.297	3.087	4.578
float	4.557	XXXXXXXX	4.620	4.725	XXXXXXXX	4.557

**Figura 12c:** CPI calculado a partir dos tempos obtidos

## 4.0 Performance Teste

```
PassMark PerformanceTest Linux

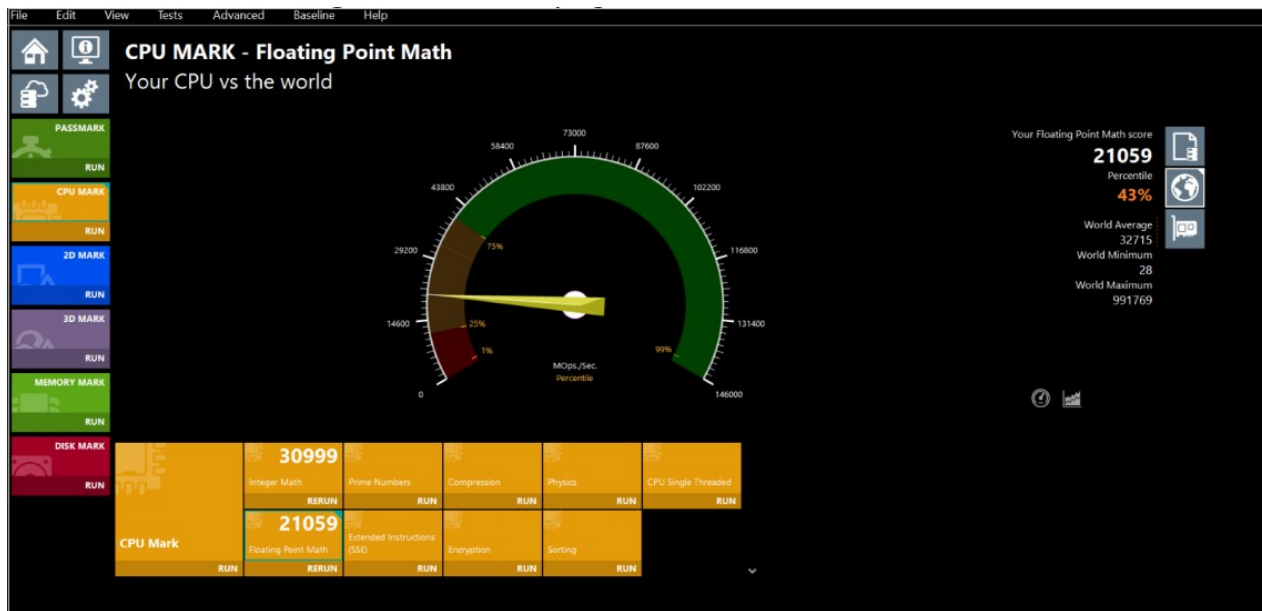
Intel Core i7-10700F CPU @ 2.90GHz (x86_64)
8 cores @ 4800 MHz | 15.5 GiB RAM
Number of Processes: 16 | Test Iterations: 1 | Test Duration: Medium
-----
CPU Mark: 15831
Integer Math 51943 Million Operations/s
Floating Point Math 30706 Million Operations/s
Prime Numbers 43.1 Million Primes/s
Sorting 31100 Thousand Strings/s
Encryption 6940 MB/s
Compression 236555 KB/s
CPU Single Threaded 2797 Million Operations/s
Physics 817 Frames/s
Extended Instructions (SSE) 12954 Million Matrices/s

Memory Mark: 2738
Database Operations 6294 Thousand Operations/s
Memory Read Cached 28195 MB/s
Memory Read Uncached 14278 MB/s
Memory Write 15269 MB/s
Available RAM 7119 Megabytes
Memory Latency 41 Nanoseconds
Memory Threaded 34657 MB/s
-----

Results not submitted
Upload results to cpubenchmark.net? [Y/n]:

Use ESC or CTRL-C to exit
A: Run All Tests C: Run CPU Tests M: Run Memory Tests U: Upload Test Results
```

Print página do Performance Teste - Frequência 2.9 GHz



Print página do Performance Teste - Frequência 2.42 GHz

```
PassMark PerformanceTest Linux

AMD Ryzen 5 5500U with Radeon Graphics (x86_64)
6 cores @ 4056 MHz | 5.7 GiB RAM
Number of Processes: 12 | Test Iterations: 1 | Test Duration: Medium
-----
CPU Mark: 12733
Integer Math 47363 Million Operations/s
Floating Point Math 21602 Million Operations/s
Prime Numbers 27.0 Million Primes/s
Sorting 18926 Thousand Strings/s
Encryption 12102 MB/s
Compression 170501 KB/s
CPU Single Threaded 2394 Million Operations/s
Physics 528 Frames/s
Extended Instructions (SSE) 9354 Million Matrices/s

Memory Mark: 1606
Database Operations 3854 Thousand Operations/s
Memory Read Cached 26877 MB/s
Memory Read Uncached 15356 MB/s
Memory Write 7579 MB/s
Available RAM 1388 Megabytes
Memory Latency 51 Nanoseconds
Memory Threaded 32518 MB/s
-----

Results not submitted
Upload results to cpubenchmark.net? [Y/n]:

Use ESC or CTRL-C to exit
A: Run All Tests C: Run CPU Tests M: Run Memory Tests U: Upload Test Results
```

Print página do Performance Teste - Frequência 2.1 GHz



## .5.0 Benchmark

	Processador	Frequência	SO	Compilador	Prog. em C		Performance Teste	
					MIPS médio (inteiros)	MFLOPS médio (FP)	MIPS (inteiros)	MFLOPS (FP)
Nilson Deon	Intel Core i5-1135G7	2.42 GHz	Windows 11	gcc 6.3.0-1	1044.4193	278.4966	30999.0000	21059.0000
Saulo de Moura	AMD Ryzen 5 5500U	2.10 GHz	Linux – Ubuntu 22.4	gcc 11.3.0	602.0181	455.1620	47363.0000	21602.0000
Gabriel Vargas	Intel Core i7-10700F	2.90 GHz	Linux – Ubuntu 22.4	gcc 11.3.0	724.3934	501.5153	51943.0000	30706.0000

Print de cada CPU com os MIPS e MFLOPS

	Identificação da máquina				Prog. em C		Performance Teste	
	Processador	Frequência	SO	Compilador	Speed up (inteiros)	Speed up (FP)	Speed up (inteiros)	Speed up (FP)
Nilson Deon	Intel Core i5-1135G7	2.42 GHz	Windows 11	gcc 6.3.0-1	1	1	1	1
Saulo de Moura	AMD Ryzen 5 5500U	2.10 GHz	Linux – Ubuntu 22.4	gcc 11.3.0	1.7349	0.6119	1.5279	1.0258
Gabriel Vargas	Intel Core i7-10700F	2.90 GHz	Linux – Ubuntu 22.4	gcc 11.3.0	1.4418	0.5553	1.6756	1.4581

Print de cada CPU com speedUp

SO	Compilador	Programa em C (inteiros)		Speed up
		Intel Core i7-10700F	AMD Ryzen 5 5500U	
Linux – Ubuntu 22.4	gcc 11.3.0	14.4800	18.61 ms	1.2852

Print do Speed Up com mesmo SO e compilador (tempos obtidos com a média de todas as operações com inteiros realizadas anteriormente)

SO	Frequência	Processador	Programa em C (inteiros)		Speed up
			Compilador gcc	Compilador g++	
Linux – Ubuntu 22.4	4056MHz	Ryzen 5 5500U	20.1 ms	17.8 ms	1.1292

Print do Speed Up trocando compilador, mas mantendo SO, Frequência e Processador

```

Tempo : 20 ms.      Tempo gasto: 18 ms.
Tempo : 20 ms.      Tempo gasto: 17 ms.
Tempo : 20 ms.      Tempo gasto: 17 ms.
Tempo : 20 ms.      Tempo gasto: 17 ms.
Tempo : 20 ms.      Tempo gasto: 18 ms.
Tempo : 20 ms.      Tempo gasto: 19 ms.
Tempo : 20 ms.      Tempo gasto: 18 ms.
Tempo : 20 ms.      Tempo gasto: 18 ms.
Tempo : 20 ms.      Tempo gasto: 18 ms.
Tempo : 20 ms.      Tempo gasto: 18 ms.
Tempo gasto media: 20 ms.  Tempo gasto media: 17.8 ms.
  
```

Compilador	Frequência	Processador	Programa em C (inteiros)		Speed up
			Windows 11	Linux Ubuntu 22.4	
Compilador gcc 11.3.0	4056MHz	Intel Core i7-10700F	15.4 ms	19.4 ms	1.2597

Print do Speed Up Compilador, Frequência e Processador, trocando SO

```

Tempo : 19 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo gasto média: 15.4 ms.

```

int: num1 = num + 3 (LINUX)

```

Tempo : 19 ms.
Tempo : 21 ms.
Tempo : 19 ms.
Tempo : 18 ms.
Tempo : 21 ms.
Tempo : 19 ms.
Tempo : 19 ms.
Tempo : 19 ms.
Tempo : 19 ms.
Tempo : 20 ms.
Tempo : 19 ms.
Tempo gasto média: 19.4 ms.

```

int: num1 = num + 3 (WINDOWS)

## 6.0 Prints Programa em C

```
gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ gcc -o main main.c
gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ ./main

char: tempo base

Tempo : 8 ms.
Tempo : 5 ms.
Tempo : 4 ms.
Tempo : 4 ms.
Tempo : 4 ms.
Tempo : 4 ms.
Tempo : 4 ms.
Tempo : 4 ms.
Tempo : 4 ms.
Tempo : 4 ms.
Tempo gasto media: 4.5 ms.
gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ gcc -o main main.c
gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ ./main

int: tempo base

Tempo : 6 ms.
Tempo : 3 ms.
Tempo : 2 ms.
Tempo : 2 ms.
Tempo : 2 ms.
Tempo : 2 ms.
Tempo : 2 ms.
Tempo : 2 ms.
Tempo : 2 ms.
Tempo : 2 ms.
Tempo : 2 ms.
Tempo gasto media: 2.5 ms.
gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ gcc -o main main.c
gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ ./main

float: tempo base

Tempo : 9 ms.
Tempo : 5 ms.
Tempo : 5 ms.
Tempo : 5 ms.
Tempo : 5 ms.
Tempo : 5 ms.
Tempo : 5 ms.
Tempo : 5 ms.
Tempo : 5 ms.
Tempo : 5 ms.
Tempo : 5 ms.
Tempo gasto media: 5.4 ms.
```

Tempos base programa em C (Linux)



```

gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ gcc -o main main.c
gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ ./main

char: num1 = num1 + 3

Tempo : 19 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo gasto media: 15.4 ms.
gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ gcc -o main main.c
gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ ./main

int: num1 = num1 + 3

Tempo : 19 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo : 15 ms.
Tempo gasto media: 15.4 ms.
gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ gcc -o main main.c
gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ ./main

float: num1 = num1 + 3

Tempo : 29 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 24 ms.
Tempo : 24 ms.
Tempo : 24 ms.
Tempo : 24 ms.
Tempo : 20 ms.
Tempo : 20 ms.
Tempo gasto media: 24 ms.

```

Tempos num1 = num1 + 3 programa em C (Linux)

```

gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ gcc -o main main.c
gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ ./main

char: num1 = num1 | num2

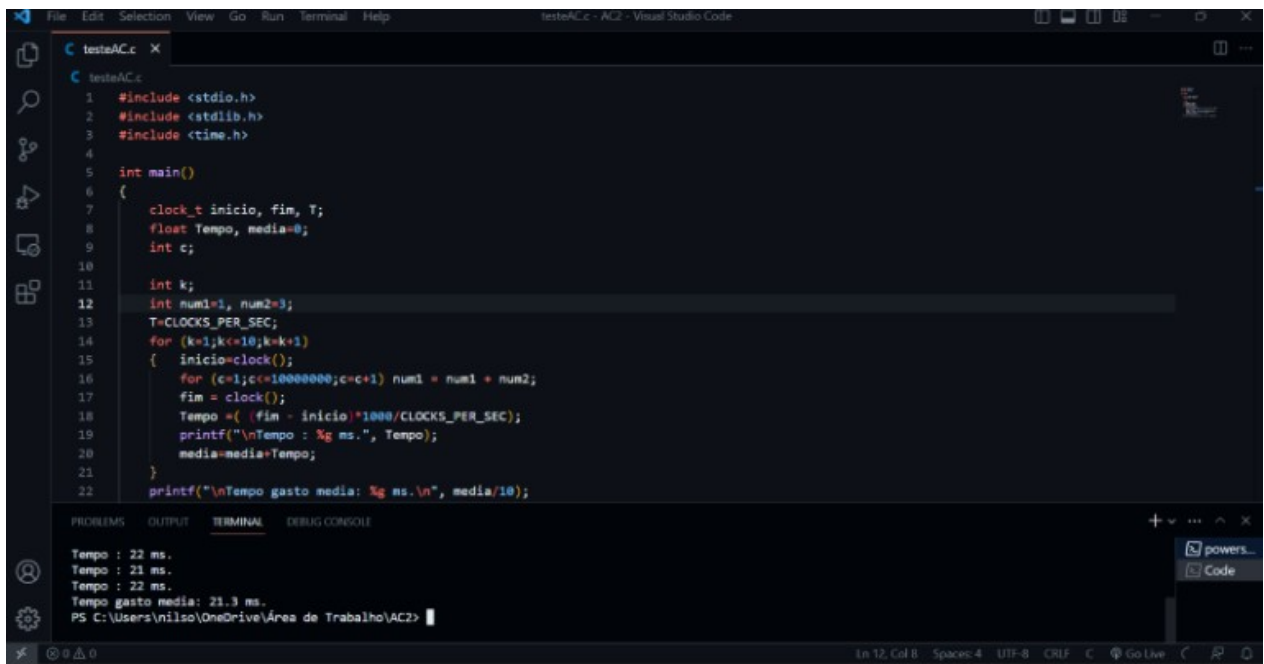
Tempo : 20 ms.
Tempo : 14 ms.
Tempo : 15 ms.
Tempo : 14 ms.
Tempo : 15 ms.
Tempo : 14 ms.
Tempo : 14 ms.
Tempo : 14 ms.
Tempo : 14 ms.
Tempo : 14 ms.
Tempo gasto media: 14.8 ms.
gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ gcc -o main main.c
gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$ ./main

int: num1 = num1 | num2

Tempo : 18 ms.
Tempo : 15 ms.
Tempo : 14 ms.
Tempo : 14 ms.
Tempo : 14 ms.
Tempo : 15 ms.
Tempo : 14 ms.
Tempo : 14 ms.
Tempo : 15 ms.
Tempo : 14 ms.
Tempo gasto media: 14.7 ms.
gabriel@desktop-ubuntu:~/Desktop/CC-PUC/3periodo/Arq2/EP04/EP04_progs/EP04_c_cpp/teste_c$

```

Tempos num1 = num1 | num2 programa em C (Linux)



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main()
6 {
7     clock_t inicio, fim, T;
8     float Tempo, media=0;
9     int c;
10
11     int k;
12     int num1=1, num2=3;
13     T=CLOCKS_PER_SEC;
14     for (k=1;k<=10;k=k+1)
15     {
16         inicio=clock();
17         for (c=1;c<=1000000;c=c+1) num1 = num1 + num2;
18         fim = clock();
19         Tempo = ( fim - inicio)*1000/CLOCKS_PER_SEC;
20         printf("\nTempo : %g ms.", Tempo);
21         media=media+Tempo;
22     }
23     printf("\nTempo gasto media: %g ms.\n", media/10);
24 }
```

Tempo : 22 ms.  
Tempo : 21 ms.  
Tempo : 22 ms.  
Tempo gasto media: 21.3 ms.  
PS C:\Users\nilso\OneDrive\Área de Trabalho\AC2>

Tempos num1 = num1 + num2 programa em C (Windows)

tempo base

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, T;
8      float Tempo, media=0;
9      int c;
10
11      int k, num1=1, num2=3;
12      T=CLOCKS_PER_SEC;
13      for (k=1;k<=10;k=k+1)
14      {  inicio=clock();
15          for (c=1;c<=10000000;c=c+1);
16          fim = clock();
17          Tempo =( (fim - inicio)*1000/CLOCKS_PER_SEC);
18          printf("\nTempo : %g ms.", Tempo);
19          media=media+Tempo;
20      }
21      printf("\nTempo gasto media: %g ms.\n", media/10);
22
23 }
```

```
Tempo : 19 ms.
Tempo : 17 ms.
Tempo : 17 ms.
Tempo : 18 ms.
Tempo : 17 ms.
Tempo : 17 ms.
Tempo : 18 ms.
Tempo : 17 ms.
Tempo : 18 ms.
Tempo : 18 ms.
Tempo gasto media: 17.6 ms.
```

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, T;
8      float Tempo, media=0;
9      int c;
10
11     int k;
12     char num1=1, num2=3;
13     T=CLOCKS_PER_SEC;
14     for (k=1;k<=10;k=k+1)
15     {
16         inicio=clock();
17         for (c=1;c<=10000000;c=c+1)num1 = num1 + 3;
18         fim = clock();
19         Tempo =( (fim - inicio)*1000/CLOCKS_PER_SEC);
20         printf("\nTempo : %g ms.", Tempo);
21         media=media+Tempo;
22     }
23     printf("\nTempo gasto media: %g ms.\n", media/10);
24 }

```

[illegible]

soma char variável:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, T;
8      float Tempo, media=0;
9      int c;
10
11     int k;
12     char num1=1, num2=3;
13     T=CLOCKS_PER_SEC;
14     for (k=1;k<=10;k=k+1)
15     {
16         inicio=clock();
17         for (c=1;c<=10000000;c=c+1)num1 = num1 + num2;
18         fim = clock();
19         Tempo =( (fim - inicio)*1000/CLOCKS_PER_SEC);
20         printf("\nTempo : %g ms.", Tempo);
21         media=media+Tempo;
22     }
23     printf("\nTempo gasto media: %g ms.\n", media/10);
24 }
```

```
Tempo : 20 ms.
Tempo : 20 ms.
Tempo : 20 ms.
Tempo : 20 ms.
Tempo : 20 ms.
Tempo : 20 ms.
Tempo : 20 ms.
Tempo : 21 ms.
Tempo : 20 ms.
Tempo : 20 ms.
Tempo gasto media: 20.1 ms.
```



```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, T;
8      float Tempo, media=0;
9      int c;
10
11     int k;
12     int num1=1, num2=3;
13     T=CLOCKS_PER_SEC;
14     for (k=1;k<=10;k=k+1)
15     {
16         inicio=clock();
17         for (c=1;c<=10000000;c=c+1) num1 = num1 + num2;
18         fim = clock();
19         Tempo =( (fim - inicio)*1000/CLOCKS_PER_SEC);
20         printf("\nTempo : %g ms.", Tempo);
21         media=media+Tempo;
22     }
23     printf("\nTempo gasto media: %g ms.\n", media/10);
24 }

```

[illegible]

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, T;
8      float Tempo, media=0;
9      int c;
10
11     int k;
12     int num1=1, num2=3;
13     T=CLOCKS_PER_SEC;
14     for (k=1;k<=10;k=k+1)
15     {
16         inicio=clock();
17         for (c=1;c<=100000000;c=c+1) num1 = num1 + num2;
18         fim = clock();
19         Tempo = ( fim - inicio)*1000/CLOCKS_PER_SEC;
20         printf("\nTempo : %g ms.", Tempo);
21         media=media+Tempo;
22     }
23     printf("\nTempo gasto media: %g ms.\n", media/10);
24 }
```

[illegible]



soma float constante:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, T;
8      float Tempo, media=0;
9      int c;
10
11     int k;
12     float num1=1, num2=3;
13     T=CLOCKS_PER_SEC;
14     for (k=1;k<=10;k=k+1)
15     {
16         inicio=clock();
17         for (c=1;c<=10000000;c=c+1)num1 = num1 + 3;
18         fim = clock();
19         Tempo =( (fim - inicio)*1000/CLOCKS_PER_SEC);
20         printf("\nTempo : %g ms.", Tempo);
21         media=media+Tempo;
22     }
23     printf("\nTempo gasto media: %g ms.\n", media/10);
24 }
```

```
Tempo : 27 ms.
Tempo : 27 ms.
Tempo : 27 ms.
Tempo : 28 ms.
Tempo : 27 ms.
Tempo : 28 ms.
Tempo : 27 ms.
Tempo : 27 ms.
Tempo : 27 ms.
Tempo : 27 ms.
Tempo : 27 ms.
Tempo gasto media: 27.2 ms.
```

Soma float variável:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, T;
8      float Tempo, media=0;
9      int c;
10
11     int k;
12     float num1=1, num2=3;
13     T=CLOCKS_PER_SEC;
14     for (k=1;k<=10;k=k+1)
15     {
16         inicio=clock();
17         for (c=1;c<=10000000;c=c+1)num1 = num1 + num2;
18         fim = clock();
19         Tempo =( (fim - inicio)*1000/CLOCKS_PER_SEC);
20         printf("\nTempo : %g ms.", Tempo);
21         media=media+Tempo;
22     }
23     printf("\nTempo gasto media: %g ms.\n", media/10);
24 }
```

```
Tempo : 34 ms.  
Tempo : 27 ms.  
Tempo : 27 ms.  
Tempo : 27 ms.  
Tempo : 27 ms.  
Tempo : 27 ms.  
Tempo : 28 ms.  
Tempo : 29 ms.  
Tempo : 27 ms.  
Tempo : 27 ms.  
Tempo gasto media: 28 ms.
```

OR char constante:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, T;
8      float Tempo, media=0;
9      int c;
10
11     int k;
12     char num1=1, num2=3;
13     T=CLOCKS_PER_SEC;
14     for (k=1;k<=10;k=k+1)
15     {
16         inicio=clock();
17         for (c=1;c<=10000000;c=c+1)num1 = num1 || 3;
18         fim = clock();
19         Tempo =( (fim - inicio)*1000/CLOCKS_PER_SEC);
20         printf("\nTempo : %g ms.", Tempo);
21         media=media+Tempo;
22     }
23     printf("\nTempo gasto media: %g ms.\n", media/10);
24 }

```

```

Tempo : 18 ms.
Tempo : 18 ms.
Tempo : 17 ms.
Tempo : 18 ms.
Tempo : 18 ms.
Tempo : 18 ms.
Tempo : 18 ms.
Tempo : 19 ms.
Tempo : 19 ms.
Tempo : 19 ms.
Tempo gasto media: 18.2 ms.

```

OR char variável:

```

2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, T;
8      float Tempo, media=0;
9      int c;
10
11     int k;
12     char num1=1, num2=3;
13     T=CLOCKS_PER_SEC;
14     for (k=1;k<=10;k=k+1)
15     {   inicio=clock();
16         for (c=1;c<=10000000;c=c+1)num1 = num1 || num2;
17         fim = clock();
18         Tempo =( (fim - inicio)*1000/CLOCKS_PER_SEC);
19         printf("\nTempo : %g ms.", Tempo);
20         media=media+Tempo;
21     }
22     printf("\nTempo gasto media: %g ms.\n", media/10);
23
24 }

```

```

Tempo : 18 ms.
Tempo : 18 ms.
Tempo : 19 ms.
Tempo : 18 ms.
Tempo : 18 ms.
Tempo : 17 ms.
Tempo : 17 ms.
Tempo : 18 ms.
Tempo : 17 ms.
Tempo : 18 ms.
Tempo gasto media: 17.8 ms.

```

OR int constante:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, T;
8      float Tempo, media=0;
9      int c;
10
11     int k;
12     int num1=1, num2=3;
13     T=CLOCKS_PER_SEC;
14     for (k=1;k<=10;k=k+1)
15     {
16         inicio=clock();
17         for (c=1;c<=10000000;c=c+1)num1 = num1 || 3;
18         fim = clock();
19         Tempo =( (fim - inicio)*1000/CLOCKS_PER_SEC);
20         printf("\nTempo : %g ms.", Tempo);
21         media=media+Tempo;
22     }
23     printf("\nTempo gasto media: %g ms.\n", media/10);
24 }

```

```

Tempo : 20 ms.
Tempo : 18 ms.
Tempo : 17 ms.
Tempo : 18 ms.
Tempo : 18 ms.
Tempo : 18 ms.
Tempo : 18 ms.
Tempo : 18 ms.
Tempo : 18 ms.
Tempo : 18 ms.
Tempo gasto media: 18.1 ms.

```

OR int variável:





```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, T;
8      float Tempo, media=0;
9      int c;
10
11     int k;
12     char num1=1, num2=3;
13     T=CLOCKS_PER_SEC;
14     for (k=1;k<=10;k=k+1)
15     {   inicio=clock();
16         for (c=1;c<=10000000;c=c+1)num1 = num1 * 3;
17         fim = clock();
18         Tempo =( fim - inicio)*1000/CLOCKS_PER_SEC);
19         printf("\nTempo : %g ms.", Tempo);
20         media=media+Tempo;
21     }
22     printf("\nTempo gasto media: %g ms.\n", media/10);
23
24 }

```

```

Tempo : 22 ms.
Tempo : 22 ms.
Tempo : 22 ms.
Tempo : 22 ms.
Tempo : 22 ms.
Tempo : 22 ms.
Tempo : 22 ms.
Tempo : 22 ms.
Tempo : 22 ms.
Tempo : 22 ms.
Tempo gasto media: 22 ms.

```

mult char variável:



```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, T;
8      float Tempo, media=0;
9      int c;
10
11     int k;
12     char num1=1, num2=3;
13     T=CLOCKS_PER_SEC;
14     for (k=1;k<=10;k=k+1)
15     {   inicio=clock();
16         for (c=1;c<=10000000;c=c+1) num1 = num1 * num2;
17         fim = clock();
18         Tempo = (fim - inicio)*1000/CLOCKS_PER_SEC;
19         printf("\nTempo : %g ms.", Tempo);
20         media=media+Tempo;
21     }
22     printf("\nTempo gasto media: %g ms.\n", media/10);
23
24 }

```

```

Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo gasto media: 25 ms.

```

mult int constante:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, T;
8      float Tempo, media=0;
9      int c;
10
11     int k;
12     int num1=1, num2=3;
13     T=CLOCKS_PER_SEC;
14     for (k=1;k<=10;k=k+1)
15     {
16         inicio=clock();
17         for (c=1;c<=10000000;c=c+1) num1 = num1 * 3;
18         fim = clock();
19         Tempo =( (fim - inicio)*1000/CLOCKS_PER_SEC);
20         printf("\nTempo : %g ms.", Tempo);
21         media=media+Tempo;
22     }
23     printf("\nTempo gasto media: %g ms.\n", media/10);
24 }

```

```

Tempo : 23 ms.
Tempo : 22 ms.
Tempo : 22 ms.
Tempo : 22 ms.
Tempo : 22 ms.
Tempo : 22 ms.
Tempo : 23 ms.
Tempo : 22 ms.
Tempo : 23 ms.
Tempo : 23 ms.
Tempo gasto media: 22.4 ms.

```

mult int variável:

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  int main()
6  {
7      clock_t inicio, fim, T;
8      float Tempo, media=0;
9      int c;
10
11     int k;
12     int num1=1, num2=3;
13     T=CLOCKS_PER_SEC;
14     for (k=1;k<=10;k=k+1)
15     {   inicio=clock();
16         for (c=1;c<=10000000;c=c+1) num1 = num1 * num2;
17         fim = clock();
18         Tempo =( (fim - inicio)*1000/CLOCKS_PER_SEC);
19         printf("\nTempo : %g ms.", Tempo);
20         media=media+Tempo;
21     }
22     printf("\nTempo gasto media: %g ms.\n", media/10);
23
24 }

```

```

Tempo : 26 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo : 25 ms.
Tempo gasto media: 25.1 ms.

```

Mult float constante:

```

int main()
{
    clock_t inicio, fim, T;
    float Tempo, media=0;
    int c;

    int k;
    float num1=1, num2=3;
    T=CLOCKS_PER_SEC;
    for (k=1;k<=10;k=k+1)
    {
        inicio=clock();
        for (c=1;c<=10000000;c=c+1) num1 = num1 * 3;
        fim = clock();
        Tempo = ( fim - inicio)*1000/CLOCKS_PER_SEC;
        printf("\nTempo : %g ms.", Tempo);
        media=media+Tempo;
    }
    printf("\nTempo gasto media: %g ms.\n", media/10);
}

```

```

Tempo : 28 ms.
Tempo : 28 ms.
Tempo : 27 ms.
Tempo : 28 ms.
Tempo : 27 ms.
Tempo : 27 ms.
Tempo : 28 ms.
Tempo : 27 ms.
Tempo : 28 ms.
Tempo : 27 ms.
Tempo gasto media: 27.5 ms.

```

Mult float variável:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main()
6 {
7     clock_t inicio, fim, T;
8     float Tempo, media=0;
9     int c;
10
11     int k;
12     float num1=1, num2=3;
13     T=CLOCKS_PER_SEC;
14     for (k=1;k<=10;k=k+1)
15     {
16         inicio=clock();
17         for (c=1;c<=10000000;c=c+1)num1 = num1 * num2;
18         fim = clock();
19         Tempo =( (fim - inicio)*1000/CLOCKS_PER_SEC);
20         printf("\nTempo : %g ms.", Tempo);
21         media=media+Tempo;
22     }
23     printf("\nTempo gasto media: %g ms.\n", media/10);
24 }
```

[illegible]