



ASSIGNMENT # 2

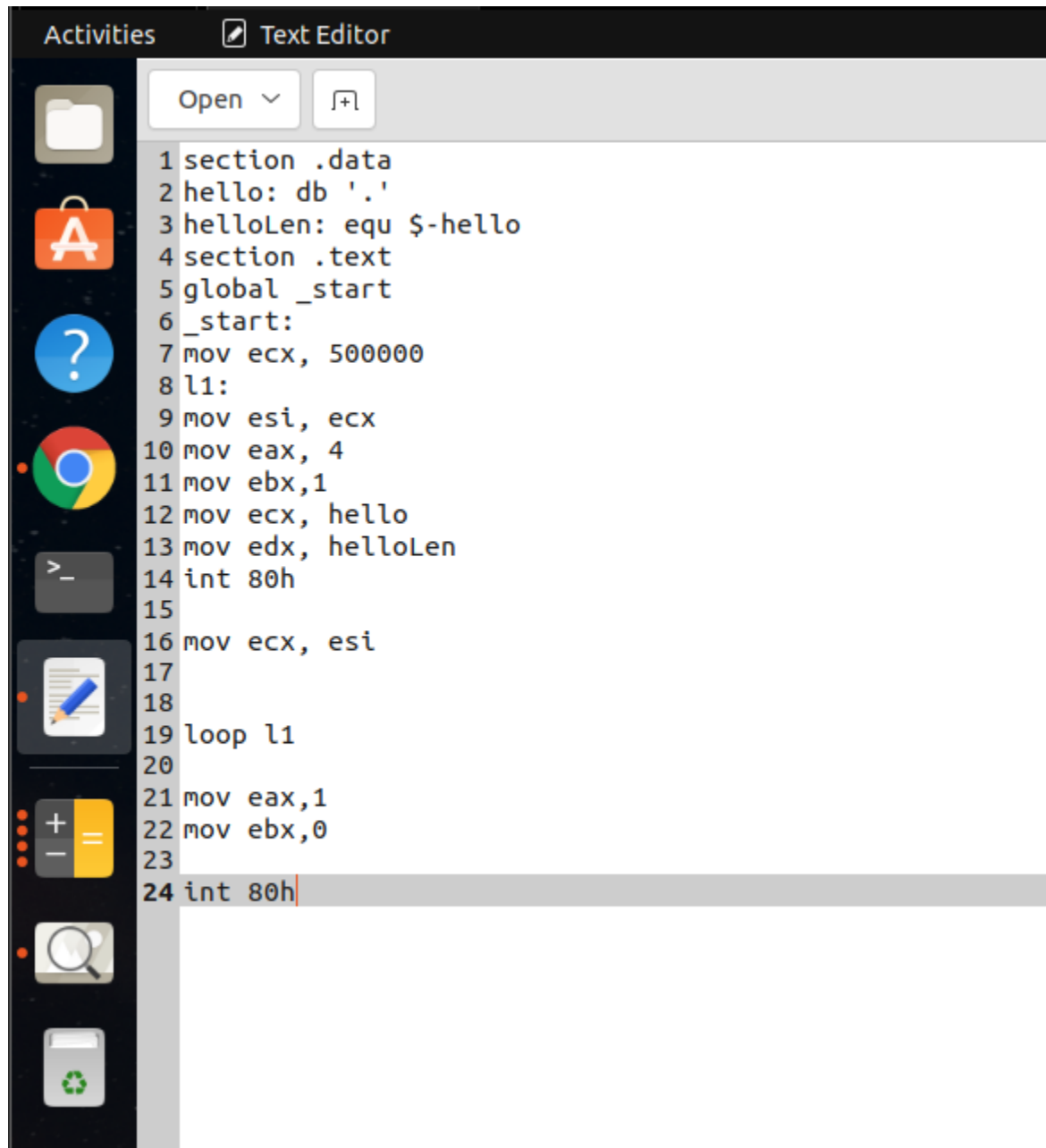
NAME : SHAHERYAR ASHFAQ

ROLL NO : **20P-0128**

SECTION : BS-CS **4-B**

SUBJECT : OPERATING SYSTEM

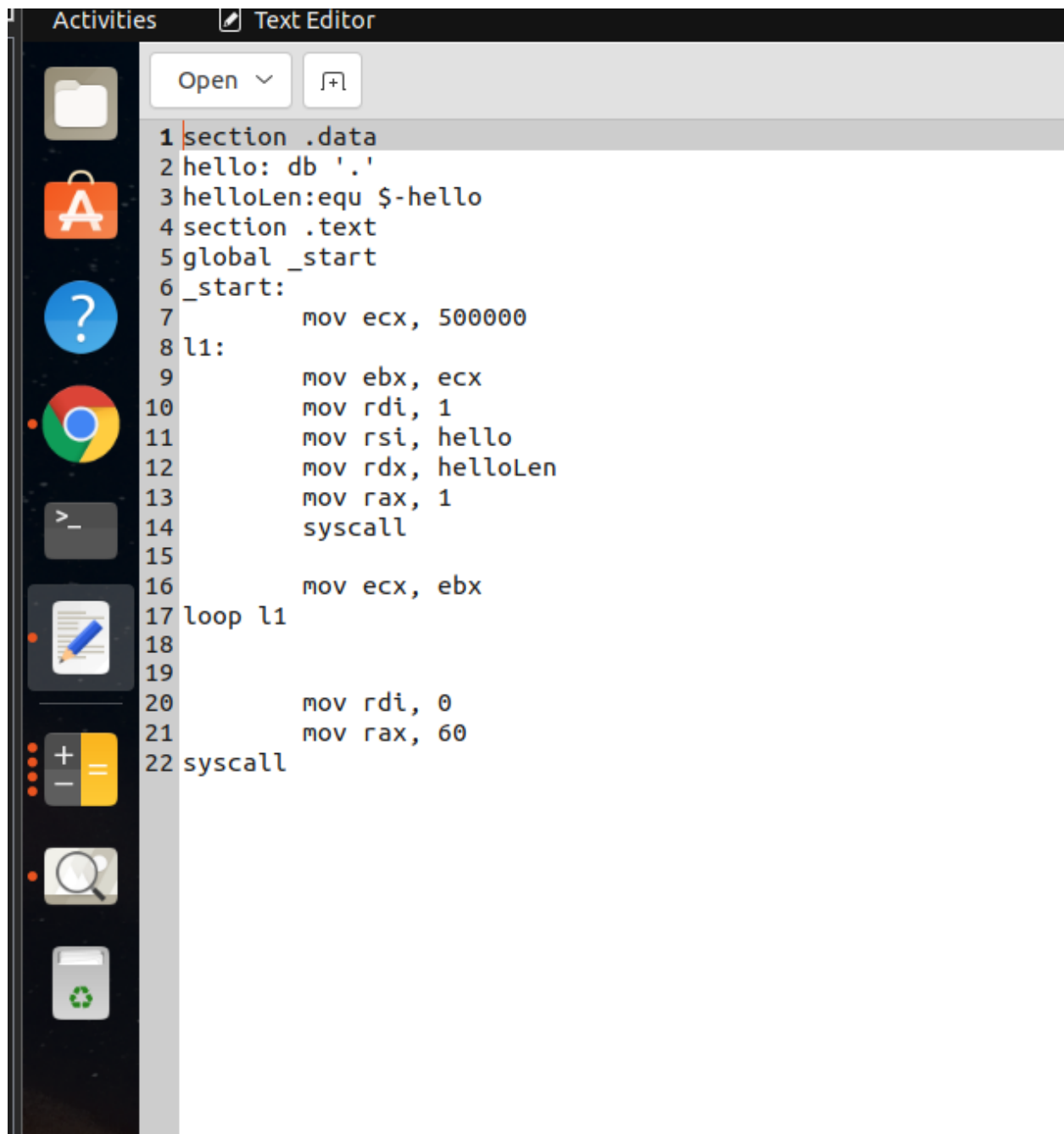
HELLO.asm



The screenshot shows a Linux desktop environment with a dark-themed sidebar on the left containing various application icons. The main window is titled 'Text Editor' and displays the assembly code for 'HELLO.asm'. The code is as follows:

```
1 section .data
2 hello: db '.'
3 helloLen: equ $-hello
4 section .text
5 global _start
6 _start:
7 mov ecx, 500000
8 l1:
9 mov esi, ecx
10 mov eax, 4
11 mov ebx, 1
12 mov ecx, hello
13 mov edx, helloLen
14 int 80h
15
16 mov ecx, esi
17
18
19 loop l1
20
21 mov eax, 1
22 mov ebx, 0
23
24 int 80h
```

HELLO2.asm





```
shaheryar@ubuntu:~/Desktop$ nasm -f elf64 hello.asm
shaheryar@ubuntu:~/Desktop$ ld -s -o hello hello.o
shaheryar@ubuntu:~/Desktop$ ./hello
```



```
eryar@ubuntu:~/Desktop$ time ./hello >/dev/null
```

```
real    0m0.299s
user    0m0.155s
sys     0m0.143s
```

```
shaheryar@ubuntu:~/Desktop$ time ./hello >/dev/null
```

```
real    0m0.300s
user    0m0.180s
sys     0m0.120s
```

```
shaheryar@ubuntu:~/Desktop$ time ./hello >/dev/null
```

```
real    0m0.298s
user    0m0.181s
sys     0m0.117s
```

```
shaheryar@ubuntu:~/Desktop$ time ./hello >/dev/null
```

```
real    0m0.298s
user    0m0.181s
sys     0m0.117s
```

```
shaheryar@ubuntu:~/Desktop$ nasm -f elf64 -o hello.o hello.asm
```

```
shaheryar@ubuntu:~/Desktop$ nasm -f elf64 -o hello2.o hello2.asm
```

```
shaheryar@ubuntu:~/Desktop$ ld -s -o hello hello.o
```

```
shaheryar@ubuntu:~/Desktop$ ld -s -o hello2 hello2.o
```

```
shaheryar@ubuntu:~/Desktop$ chmod +x runner.sh
```

```
shaheryar@ubuntu:~/Desktop$ ./runner.sh
```

```
shaheryar@ubuntu:~/Desktop$
```



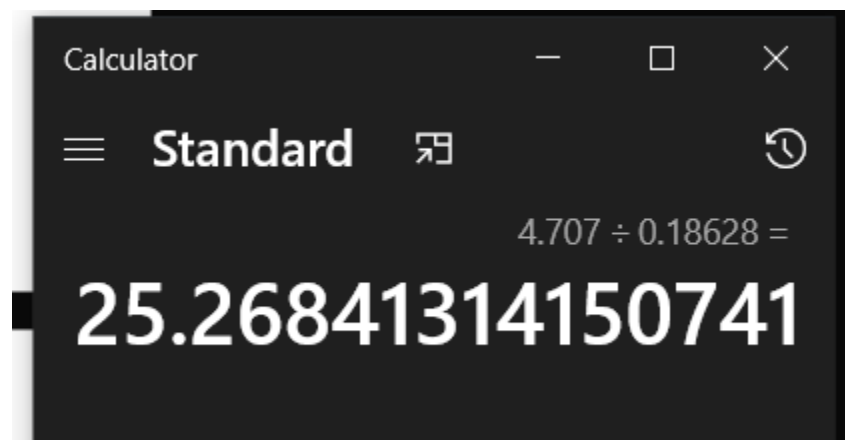
Attempt 01

Number of experiments run: N = 50

Average 'user time' for hello (int-based calls) I = **0.18628**

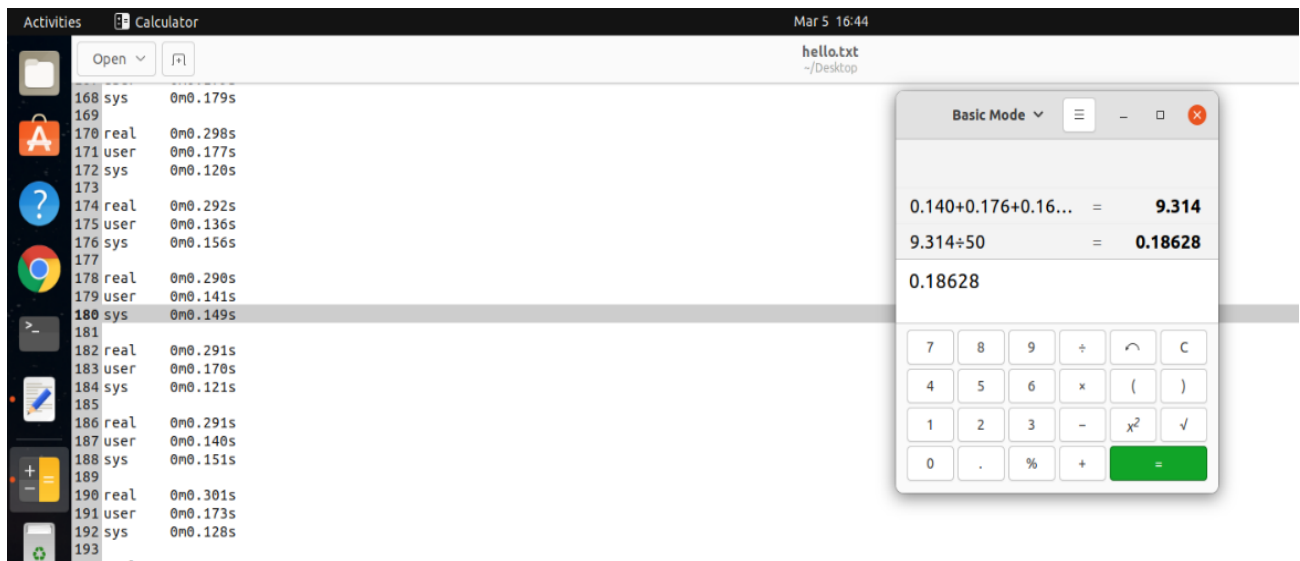
Average 'user time' for hello2 (syscall-based calls) S = **0.13916**

Percentage speedup: $(I-S)*100/I =$ **25.26841314150741**



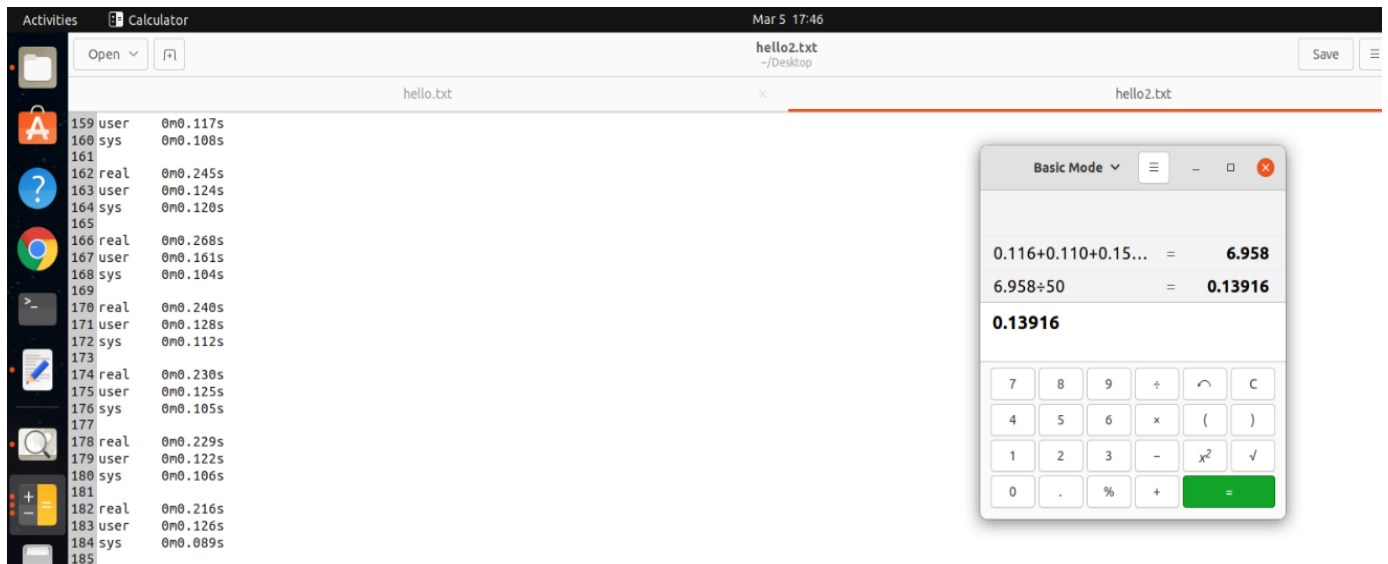
INT BASED CALL

$(0.140+0.176+0.164+0.156+0.195+0.210+0.188+0.159+0.210+0.252+0.125+0.187+0.200+0.230+0.166+0.155+0.264+0.221+0.209+0.196+0.172+0.185+0.230+0.192+0.185+0.187+0.156+0.185+0.340+0.184+0.146+0.204+0.197+0.181+0.156+0.165+0.199+0.176+0.148+0.161+0.233+0.179+0.177+0.136+0.141+0.170+0.140+0.173+0.213+0.200)/50=$ **0.18628**



SYSCALL-BASED CALLS

$(0.116 + 0.110 + 0.152 + 0.149 + 0.158 + 0.116 + 0.101 + 0.133 + 0.266 + 0.160 + 0.140 + 0.157 + 0.132 + 0.118 + 0.145 + 0.168 + 0.208 + 0.149 + 0.136 + 0.128 + 0.144 + 0.131 + 0.163 + 0.117 + 0.116 + 0.131 + 0.120 + 0.308 + 0.134 + 0.106 + 0.094 + 0.171 + 0.145 + 0.151 + 0.104 + 0.131 + 0.144 + 0.148 + 0.117 + 0.124 + 0.161 + 0.128 + 0.125 + 0.122 + 0.126 + 0.118 + 0.129 + 0.136 + 0.172) / 50 = 0.13916$



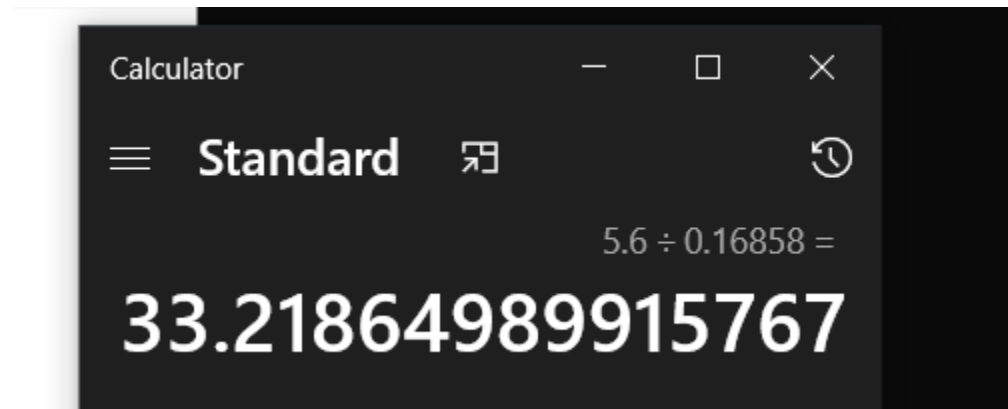
Attempt 02

Number of experiments run: N = 50

Average 'user time' for hello (int-based calls) I = **0.16858**

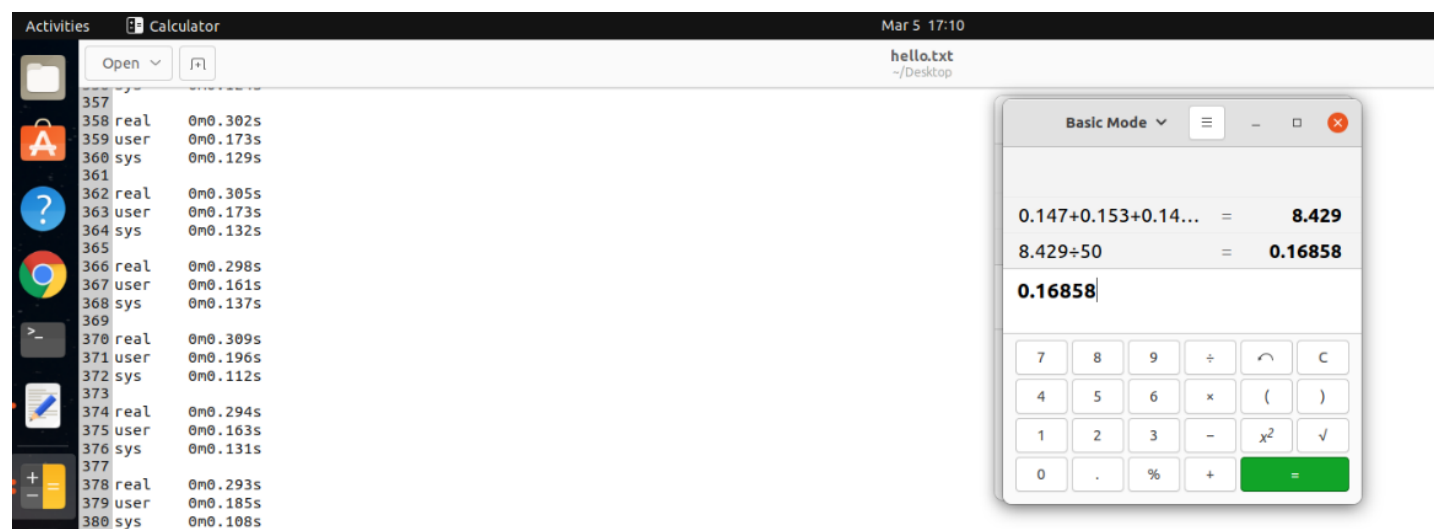
Average 'user time' for hello2 (syscall-based calls) S = **0.11258**

Percentage speedup: $(I-S)*100/I = 33.21864989915767$



INT BASED CALL

$(0.147+0.153+0.145+0.173+0.159+0.136+0.156+0.174+0.160+0.156+0.191+0.171+0.136+0.172+0.193+0.169+0.187+0.141+0.148+0.254+0.164+0.176+0.152+0.160+0.165+0.144+0.165+0.196+0.164+0.164+0.1760+0.201+0.335+0.200+0.144+0.163+0.173+0.180+0.173+0.173+0.161+0.196+0.163+0.185+0.158+0.163+0.172+0.168+0.174)/50 = \mathbf{0.16858}$



SYSCALL-BASED CALLS

$(0.141+0.128+0.128+0.148+0.122+0.131+0.105+0.124+0.109+0.116+0.112+0.096+0.112+0.148+0.166+0.092+0.097+0.120+0.084+0.154+0.092+0.116+0.103+0.111+0.104+0.134+0.127+0.155+0.134+0.122+0.112+0.221+0.136+0.011+0.110+0.141+0.138+0.088+0.143+0.092+0.120+0.116+0.106+0.134+0.096+0.116+0.118)/50 = \mathbf{0.11258}$

