# Roll NO:20BCE075

In

[1]:

In

[2]:

total no of element in array for loop 1 : 5

Enter Elements :

**import**

time

num\_elements

**=**

int

(

input

(

"total no of element in array for loop 1 : "

))

user\_data

**=**

[]

print

(

"Enter Elements :"

)

**for**

i

**in**

range

(

num\_elements

):

element

**=**

int

(

input

())

user\_data

.

append

(

element

)

12

13

14

15

16

# Loop unrolling

In

[3]:

**def**

original\_loop

(

arr

):

sum

**=**

0

**for**

i

**in**

range

(

len

(

arr

)):

sum

**+=**

arr

[

i

]

**return**

sum

start

**=**

time

.

time

()

**for**

i

**in**

range

(

100000

):

original\_loop

(

user\_data

)

end

**=**

time

.

time

()

original\_time

**=**

(

end

**-**

start

)

**\***

1

e

6

*# Convert to microseconds*

print

(

f"\nTime for Original Loop:

{

original\_time

}

microseconds"

)

Time for Original Loop: 192578.07731628418 microseconds

In

[4]:

**def**

loop\_unrolling

(

arr

):

sum

**=**

0

length

**=**

len

(

arr

)

i

**=**

0

**while**

i

**+**

4

**<=**

length

:

sum

**+=**

arr

[

i

]

sum

**+=**

arr

[

i

**+**

1

]

sum

**+=**

arr

[

i

**+**

2

]

sum

**+=**

arr

[

i

**+**

3

]

i

**+=**

4

**while**

i

**<**

length

:

sum

**+=**

arr

[

i

]

i

**+=**

1

**return**

sum

start

**=**

time

.

time

()

**for**

i

**in**

range

(

100000

):

loop\_unrolling

(

user\_data

)

end

**=**

time

.

time

()

unrolled\_time

**=**

(

end

**-**

start

)

**\***

1

e

6

*# Convert to microseconds*

print

(

f"Time after Loop Unrolling:

{

unrolled\_time

}

microseconds\n"

)

Time after Loop Unrolling: 186233.5205078125 microseconds

In

[5]:

In

[6]:

size

**=**

100000

A

**=**

[

i

**for**

i

**in**

range

(

size

)]

B

**=**

[

size

**-**

i

**for**

i

**in**

range

(

size

)]

C

**=**

[

0

]

**\***

size

D

**=**

[

0

]

**\***

size

start\_time

**=**

time

.

time

()

**for**

i

**in**

range

(

size

):

C

[

i

]

**=**

A

[

i

]

**+**

B

[

i

]

**for**

i

**in**

range

(

size

):

D

[

i

]

**=**

A

[

i

]

**-**

B

[

i

]

end\_time

**=**

time

.

time

()

duration

**=**

(

end\_time

**-**

start\_time

)

**\***

1

e

6

*# Convert to microseconds*

print

(

f"Original Code Execution Time:

{

duration

}

microseconds"

)

Original Code Execution Time: 106626.51062011719 microseconds

In

[7]:

start\_time

**=**

time

.

time

()

**for**

i

**in**

range

(

size

):

C

[

i

]

**=**

A

[

i

]

**+**

B

[

i

]

D

[

i

]

**=**

A

[

i

]

**-**

B

[

i

]

end\_time

**=**

time

.

time

()

duration

**=**

(

end\_time

**-**

start\_time

)

**\***

1

e

6

*# Convert to microseconds*

print

(

f"Optimized (Fused) Code Execution Time:

{

duration

}

microseconds"

)

**for**

i

**in**

range

(

size

):

**if**

C

[

i

]

**!=**

A

[

i

]

**+**

B

[

i

]

**or**

D

[

i

]

**!=**

A

[

i

]

**-**

B

[

i

]:

print

(

"Error: Results do not match."

)

exit

(

1

)

exit

(

0

)

Optimized (Fused) Code Execution Time: 98369.12155151367 microseconds In [ ]: