Task name: Segregate positive and negative integers

Task no.: 9

	Name	ID
1	عامر مدحت احمد محمد	20210484
2	ماريا صفوت داود عبدالله	20210708
3	مرفت محمد حسن احمد	20210892
4	مريم احمد صابر محمد	20210904
5	مصطفى محمد سعد الطويل	20210943
6	محمد اسماعيل حسن محمد	20210741

<u>Pseudo code for the non–recursive function:</u>

Alg separatePosNNeg(A[0.....n-1]) // non_recursive

    for i ⟵ 0 to n-1

        do if (A[i]>0 && A[i+1]<0)

            then tmp=A[i]

                A[i]=A[i+1]

                A[i+1]= tmp

                i=i-2

<u>Analysis:</u>

for loop: $\sum_{i=0}^{n-1} 1$ = n-1-0+1 = n

if condition: c/1

Time complexity: Θ(n)

Output:

```
"E:\Level 2 (2020-2021)\Algorithm\algTask9_groupNo.81\alg9_nonRe.exe"

Enter your array enter
9
-3
5
-2
-8
-6
1
3
-3      -2      -8      -6      9       5       1       3
Process returned 0 (0x0)   execution time : 27.246 s
Press any key to continue.
```

```
"E:\Level 2 (2020-2021)\Algorithm\algTask9_groupNo.81\alg9_nonRe.exe"

Enter your array enter
1
-3
9
-20
4
2
-1
-8
-3      -20     -1      -8      1       9       4       2
Process returned 0 (0x0)   execution time : 20.626 s
Press any key to continue.
```

Pseudo code for the recursive function:

```
Func separatePosNeg( A[0....1], l , r){

  if (l>r)

      then return;

  if (a[l]>0 && a[l+1]<0 && l>=0){

    tmp=a[l];

    a[l]=a[l+1];

    a[l+1]=tmp;

    separatePosNeg(a, l-1, r);

  }

  else {

    separatePosNeg(a, l+1, r);

  }

}
```

<u>Analysis:</u>

if_else statement: $T_{if}/T_{else} = \max(T_{if}, T_{else})$

1st if: c/1

2nd if: $T(n) = T(n-1) + c/1$ (say 1)   k=1   (by iteration)

$T(n-1) = T(n-2) + 1$

$T(n) = T(n-2) + 2$ k=2

$T(n-2) = T(n-3) + 1$

$T(n) = T(n-3) + 3$ k=3

So, $T(n) = T(n-k) + k$ n-k=1   so, k=n-1

$T(n) = n$

Time complexity: $\Theta(n)$

else: $T(n) = T(n-1)$ k=1   **(by iteration)**

$T(n-1) = T(n-2)$

$T(n) = T(n-2)$ k=2
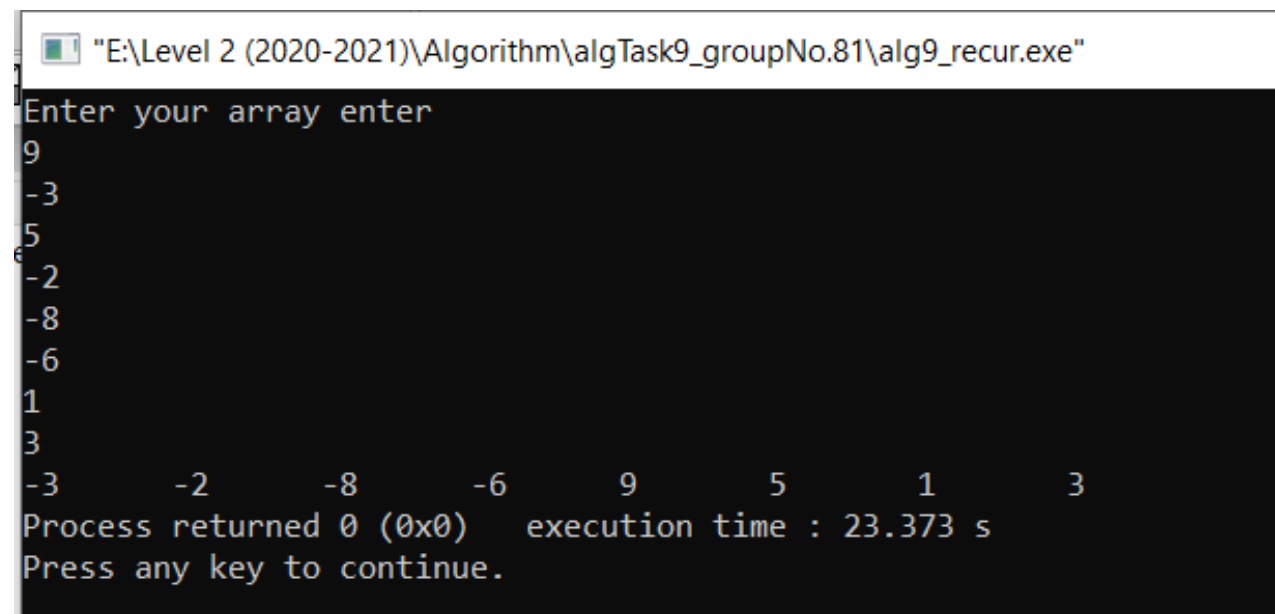
$T(n-2) = T(n-3)$

$T(n) = T(n-3)$ k=3

So, $T(n) = T(n-k) + k$ n-k=1   so, k=n-1

$T(n) = n$

Time complexity: $\Theta(n)$

So, best case will be $\Omega (n)$ as it will always runs else and not the second if condition (such as all positive values first sorted before the negative values). But (if was the opposite all positive values first then all negative values) worst case will be $O(n^2)$ as it will always runs the second if condition then runs else afterwards.

Output:

"E:\Level 2 (2020-2021)\Algorithm\algTask9_groupNo.81\alg9_recur.exe"

```
Enter your array enter
9
-3
5
-2
-8
-6
1
3
-3      -2      -8      -6      9       5       1       3
Process returned 0 (0x0)   execution time : 23.373 s
Press any key to continue.
```

Another non-recursive function:

Alg separatePosNNeg(A[0.....n-1]) // another non_recursive

 while(++i<MAXSIZE){

   if (a[i-1] > 0 && a[i] < 0)

    do tmp= a[i-1];

      a[i-1]=a[i];

      a[i]=tmp;

      i-=2;

  }

Analysis:

While loop (basic operation (increment/decrement)) so,

Time complexity: Ө (n).

Output:

"E:\Level 2 (2020-2021)\Algorithm\algTask9_groupNo.81\alg9_nonRe.exe"

```
Enter your array enter
9
-3
5
-2
-8
-6
1
3
-3      -2      -8      -6      9       5       1       3
Process returned 0 (0x0)   execution time : 25.950 s
Press any key to continue.
```

```
"E:\Level 2 (2020-2021)\Algorithm\algTask9_groupNo.81\alg9_nonRe.exe"

Enter your array enter
1
-3
9
-20
4
2
-1
-8
-3      -20     -1      -8      1       9       4       2
Process returned 0 (0x0)   execution time : 22.241 s
Press any key to continue.
```

Comparison:

	Best	worst	stable	Serial/ parallel	Memory usage
Non-recursive	n	n	Yes	serial	In-place
Recursive	n	n^2	Yes	serial	In-place

The most efficient code according to the time complexity is the non-recursive (either for loop or while loop) as best case same as worst case Θ (n).