Practical-3

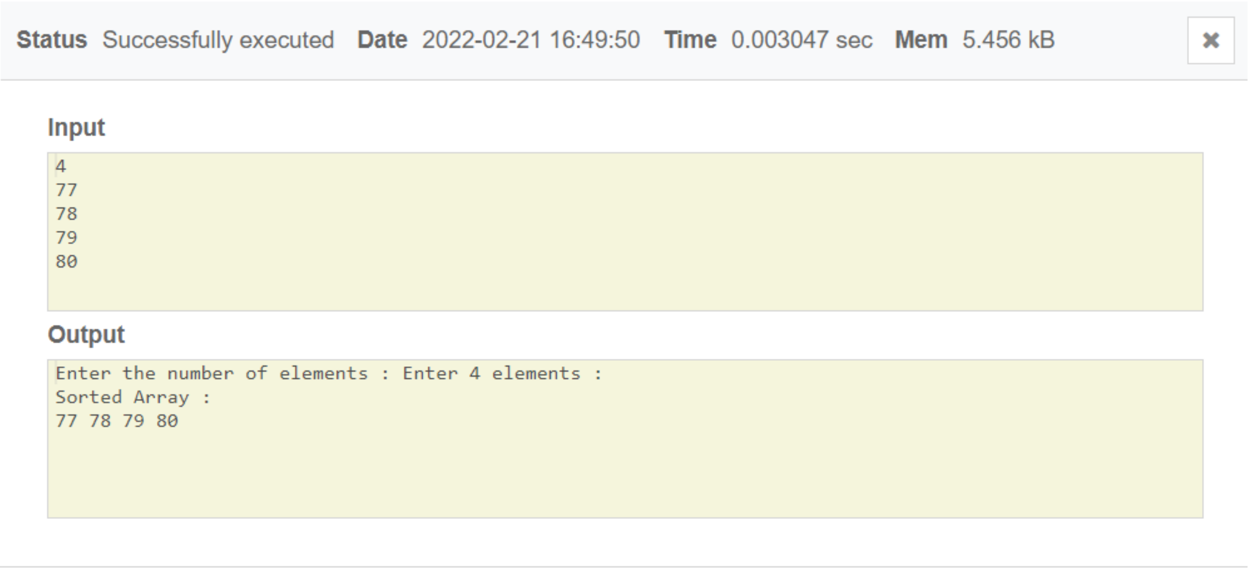
Implementation of various sorting methods(bubble and selection) and performing their timing analysis.

1. BUBBLE SORT

Code:

```
#include <stdio.h>

void swap(int *a, int *b)
{ int temp =
*a; *a = *b;
*b = temp;
}

void Bubble_Sort(int arr[], int n)
{
int i, j;
for(i = 0; i<n; i++) for (j
= 0; j<n-i-1; j++) if
(arr[j] > arr[j+1])
swap(&arr[j], &arr[j+1]);

}

void Display(int arr[], int size)
{
for(int i=0; i<size; i++)
printf("%d ", arr[i]);

}

int main()
{
int n; printf("Enter the number of elements of
array: "); scanf("%d",&n);
int arr[n];
for(int i=0; i<n; i++)
{ printf("Enter the %d element:
",i+1); scanf("%d",&arr[i]); }
n = sizeof(arr)/sizeof(arr[0]);
Bubble_Sort(arr, n);

printf("\nSorted array: ");
Display(arr, n);

return 0;
}
```

OUTPUT:



Status Successfully executed **Date** 2022-02-21 16:49:50 **Time** 0.003047 sec **Mem** 5.456 kB

Input

```
4
77
78
79
80
```

Output

```
Enter the number of elements : Enter 4 elements :
Sorted Array :
77 78 79 80
```

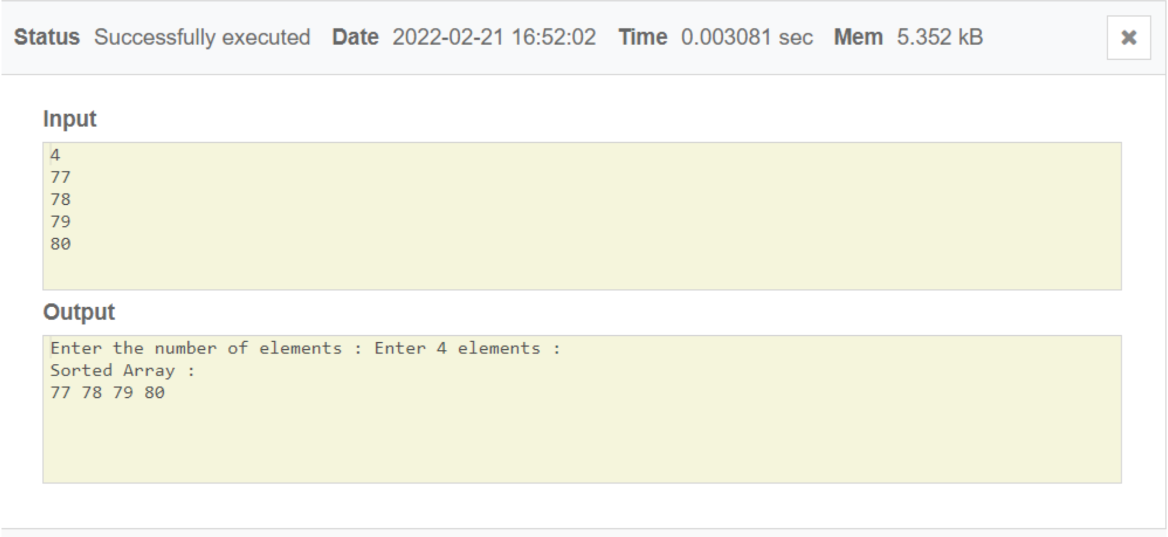
2. SELECTION SORT

Code:

```
#include <stdio.h>

void swap(int *a, int *b)
{ int temp = *a; *a =
*b;
*b = temp;
}
```

```
void Selection_Sort(int arr[], int n)
{
int i, j, min_idx;

for(i = 0; i<n-1; i++)
{ min_idx = i; for(j =
i+1; j<n; j++) if (arr[j]
< arr[min_idx])
min_idx = j;

swap(&arr[min_idx], &arr[i]);
}
}

void Display(int arr[], int size)
{
int i; for (i=0; i <
size; i++) printf("%d
", arr[i]);

}

int main()
{
int n; printf("Enter the number of elemnts of
array: "); scanf("%d",&n);
int arr[n];

for(int i=0; i<n; i++)
{ printf("Enter the %d element:
",i+1); scanf("%d",&arr[i]); }

n = sizeof(arr)/sizeof(arr[0]);
Selection_Sort(arr, n);

printf("\nSorted array: ");
Display(arr, n);
return 0;
}
```

OUTPUT:



Status Successfully executed Date 2022-02-21 16:52:02 Time 0.003081 sec Mem 5.352 kB

Input

```
4
77
78
79
80
```

Output

```
Enter the number of elements : Enter 4 elements :
Sorted Array :
77 78 79 80
```