11/11/24, 1:42 AM Bucket_Sort.c

SEM 3\Exp9\Bucket_Sort.c

```
#include <stdio.h>
2
   #include <stdlib.h>
 3
 4
   #define BUCKET_SIZE 10
 5
   typedef struct Bucket
 6
 7
 8
        int count;
 9
        int *values;
10
   } Bucket;
11
    void bucketSort(int array[], int n);
12
13
    void insertionSort(int array[], int n);
14
15
   void bucketSort(int array[], int n)
16
17
        Bucket buckets[BUCKET_SIZE];
18
        for (int i = 0; i < BUCKET_SIZE; i++)</pre>
19
20
            buckets[i].count = 0;
            buckets[i].values = (int *)malloc(n * sizeof(int));
21
22
        }
23
        for (int i = 0; i < n; i++)</pre>
24
25
26
            int bucketIndex = array[i] / BUCKET_SIZE;
27
            buckets[bucketIndex].values[buckets[bucketIndex].count++] = array[i];
        }
28
29
30
        for (int i = 0; i < BUCKET_SIZE; i++)</pre>
31
32
            if (buckets[i].count > 0)
33
34
                 insertionSort(buckets[i].values, buckets[i].count);
35
            }
        }
36
37
38
        int index = 0;
39
        for (int i = 0; i < BUCKET_SIZE; i++)</pre>
40
41
            for (int j = 0; j < buckets[i].count; j++)</pre>
42
                 array[index++] = buckets[i].values[j];
43
44
45
            free(buckets[i].values);
        }
46
47
48
49
   void insertionSort(int array[], int n)
50
51
        for (int i = 1; i < n; i++)</pre>
```

```
11/11/24, 1:42 AM
```

```
52
        {
            int key = array[i];
53
            int j = i - 1;
54
            while (j >= 0 && array[j] > key)
55
56
                array[j + 1] = array[j];
57
58
59
            array[j + 1] = key;
60
61
        }
62 }
63
```