

	<p>Mühendislik Fakültesi</p> <p>Bilgisayar Mühendisliği Programı</p> <p>BLM307E Algorithm Analysis & Design Lab Work</p>	<p>Adı Soyadı:</p> <p>Notu:</p>
---	---	---------------------------------

Lab 7: Hidden Sort

Question 1: Below, pseudo-code for hidden sort is given:
Implement the hidden sort algorithm using this pseudo-code.

```

Let C[0..k] be a new array
for i = 0 to k
    C[i] = 0
for j = 1 to A.length
    C[A[j]] = C[A[j]] + 1
for i = 1 to k
    C[i] = C[i] + C[i - 1]
for j = A.length to 1
    B[C[A[j]]] = A[j]
    C[A[j]] = C[A[j]] - 1


```

Void HiddenSort(int a[],int k,int n)

```

{
    int b[n],c[k+1];
    for(int i=0;i<=k;i++)
        c[i]=0;
    for(int i=0;i<n;i++)
        c[a[i]]++;
    for(int i=1;i<=k;i++)
        c[i]=c[i]+c[i-1];
    for(int i=n-1;i>=0;i--)
    {
        b[c[a[i]]-1]=a[i];
        c[a[i]]-=1;
    }
    for(int i=0;i<n;i++)
        a[i]=b[i];
}

```

	<p>Mühendislik Fakültesi</p> <p>Bilgisayar Mühendisliği Programı</p> <p>BLM307E Algorithm Analysis & Design Lab Work</p>	<p>Adı Soyadı:</p> <p>Notu:</p>
---	---	---------------------------------

Question 2: Assume that we have infinite memory. Which algorithm shows better performance in the worst case? QuickSort, MergeSort, Bubble Sort, Insertion Sort, Hidden Sort, Heap Sort? Why? Explain and compare time complexity.

Quicksort: $O(N^2)$

Merge Sort : $O(N \log N)$

Bubble Sort : $O(N^2)$

Insertion Sort: $O(N^2)$

Hidden Sort: $O(N + k)$

Heap Sort : $(N \log N)$

The quick sort produces the most effective and widely used method of sorting a list of any item size.

Question 3: What is the disadvantage of the hidden sort algorithm? Can we sort any data type with Hidden sort algorithm? Explain.

Hidden sort can not be used for array with non integer elements. It is easy to sort the array elements, but we have to maintain an auxiliary array to get the occurrence of the unique elements from the input array; once you get the basic idea and understand of its internal working, then it is easy to implement and use. inefficient if the range of key value (k) is very large.

Question 4: Modify your implementation so that your Hidden Sort method can sort the negative values.

```

for (i=0;i<=max;i++){
    C[i]=0;
}

for (i=0;i<n;i++){
    C[vet[i]-min]++;
}

for (i=1;i<=(max-min);i++){
    C[i]=C[i]+C[i-1];
}

for (i=n-1;i>=0;i--){
    B[C[(vet[i]-min)-1]]=vet[i];
    C[vet[i]-min]--;
}

for (i=0;i<n;i++){
    printf("%d ",B[i]);
}

```