**Binary Search**

#include <stdio.h>

int BinarySearch(int arr[], int low, int high, int x)

{

if (high >= low)

{

int mid = low + (high - low) / 2;

if (arr[mid] == x)

return mid;

if (arr[mid] > x)

return BinarySearch(arr, low, mid - 1, x);

return BinarySearch(arr, mid + 1, high, x);

}

return -1;

}

int main()

{

int size;

puts("Enter the size of the element : ");

scanf("%d", &size);

int arr[size];

for (int i = 0; i < size; ++i)

{

printf("Enter the %d element : ", i + 1);

scanf("%d", &arr[i]);

}

int x;

puts("Enter the element to search for : ");

scanf("%d", &x);

int result = BinarySearch(arr, 0, size - 1, x);

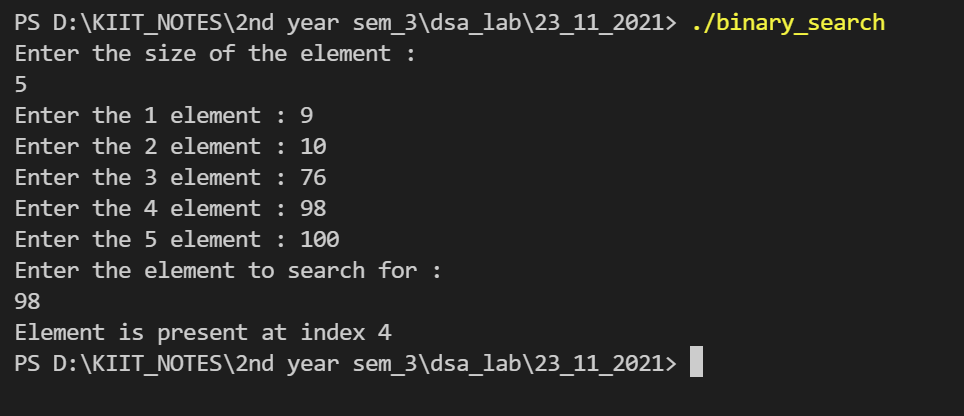
(result == -1)

? printf("Element is not present ")

: printf("Element is present at index %d", result+1);

return 0;

}



**Linear Search**

#include <stdio.h>

int Linearsearch(int arr[], int size, int x)

{

for (int i = 0; i < size; i++)

if (arr[i] == x)

return i;

return -1;

}

int main()

{

int size;

puts("Enter the size of the element : ");

scanf("%d", &size);

int arr[size];

for (int i = 0; i < size; ++i)

{

printf("Enter the %d element : ", i + 1);

scanf("%d", &arr[i]);

}

int x;

puts("Enter the element to search for : ");

scanf("%d", &x);

int result = Linearsearch(arr, size, x);

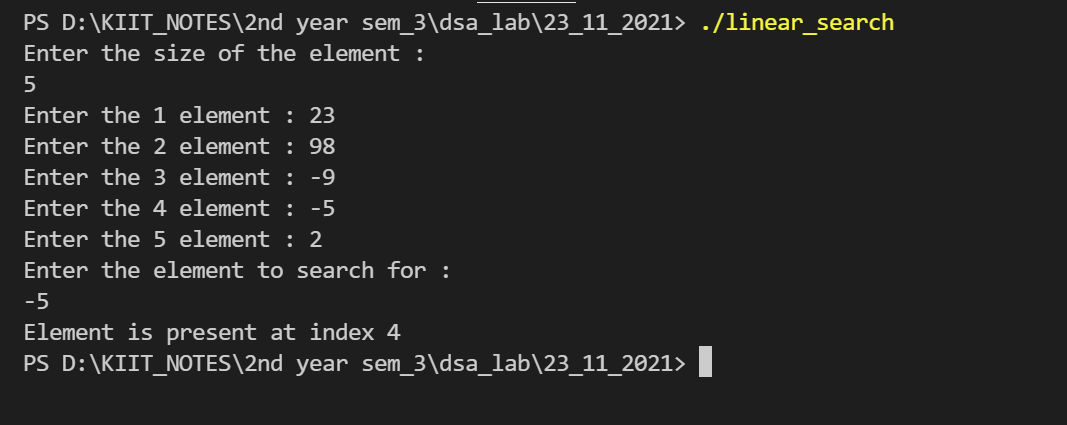
(result == -1)

? printf("Element is not present ")

: printf("Element is present at index %d", result+1);

return 0;

}



**Heap Sort**

#include <stdio.h>

void heapify(int a[], int n, int i)

{

int largest = i;

int left = 2 \* i + 1;

int right = 2 \* i + 2;

if (left < n && a[left] > a[largest])

largest = left;

if (right < n && a[right] > a[largest])

largest = right;

if (largest != i)

{

int temp = a[i];

a[i] = a[largest];

a[largest] = temp;

heapify(a, n, largest);

}

}

void heapSort(int a[], int n)

{

for (int i = n / 2 - 1; i >= 0; i--)

heapify(a, n, i);

for (int i = n - 1; i >= 0; i--)

{

int temp = a[0];

a[0] = a[i];

a[i] = temp;

heapify(a, i, 0);

}

}

int main()

{

int count = 0;

printf("Enter the number of elements : ");

scanf("%d", &count);

int a[count];

for (int i = 0; i < count; i++)

{

printf("Enter %d element : ", i + 1);

scanf("%d", &a[i]);

}

heapSort(a, count);

printf("\nDisplaying Sorted array in ascending order: \n");

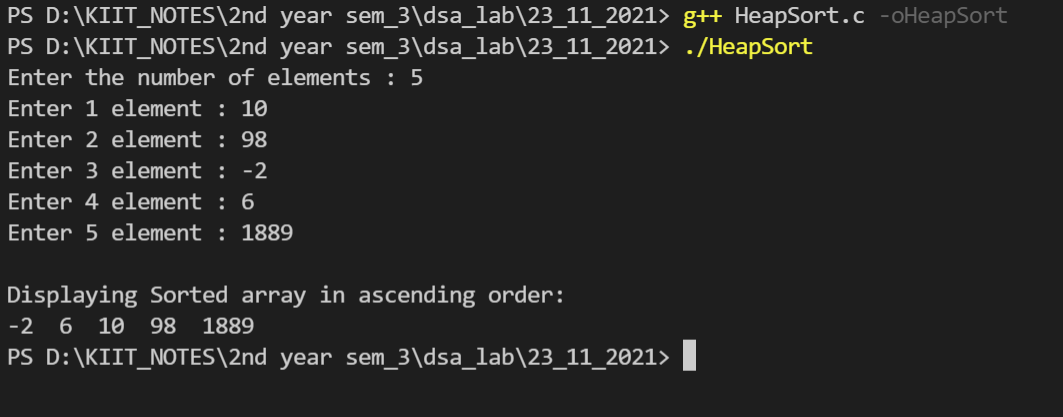
for (int i = 0; i < count; ++i)

printf("%d ", a[i]);

printf("\n");

return 0;

}



**Quick Sort**

#include <stdio.h>

#include <stdlib.h>

int \*arr;

int n;

void display()

{

printf("Elements of the array : \n");

for (int i = 0; i < n; i++)

{

printf("%d ", arr[i]);

}

printf("\n");

}

int partition(int \*A, int low, int high)

{

int pivot = A[low];

int i = low + 1;

int j = high;

int temp;

do

{

while (A[i] <= pivot)

{

i++;

}

while (A[j] > pivot)

{

j--;

}

if (i < j)

{

temp = A[i];

A[i] = A[j];

A[j] = temp;

}

} while (i < j);

temp = A[low];

A[low] = A[j];

A[j] = temp;

return j;

}

void quickSort(int \*A, int low, int high)

{

int partitionIndex;

if (low < high)

{

partitionIndex = partition(A, low, high);

quickSort(A, low, partitionIndex - 1);

quickSort(A, partitionIndex + 1, high);

}

}

int main()

{

printf("Enter the size of the array : \n");

scanf("%d", &n);

arr = (int \*)malloc(n \* sizeof(int));

for (int i = 0; i < n; i++)

{

printf("Enter %d element : \n", i + 1);

scanf("%d", &arr[i]);

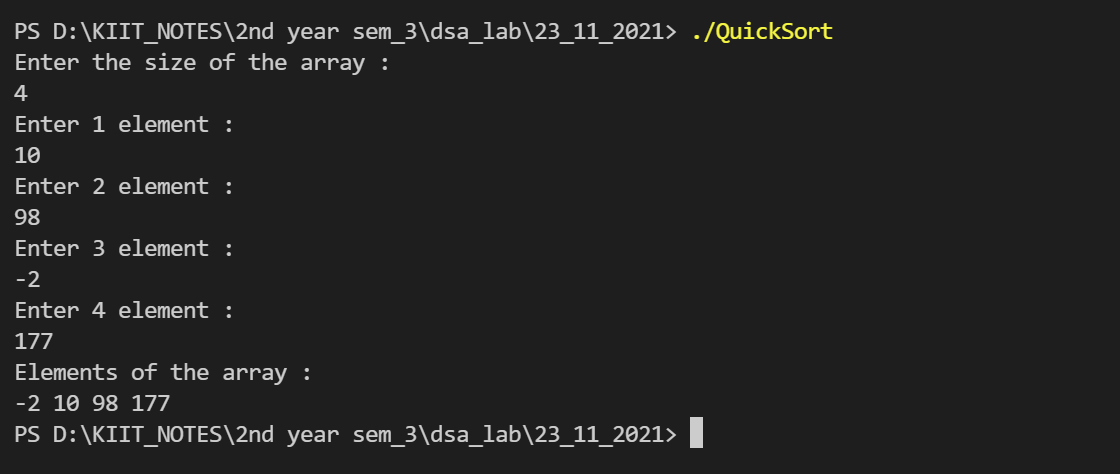
}

quickSort(arr, 0, n - 1);

display();

return 0;

}



**Radix Sort**

#include <stdio.h>

int getMax(int a[], int n)

{

int max = a[0];

for (int i = 1; i < n; i++)

{

if (a[i] > max)

max = a[i];

}

return max;

}

void countingSort(int a[], int n, int place)

{

int output[n + 1];

int count[10] = {0};

for (int i = 0; i < n; i++)

count[(a[i] / place) % 10]++;

for (int i = 1; i < 10; i++)

count[i] += count[i - 1];

for (int i = n - 1; i >= 0; i--)

{

output[count[(a[i] / place) % 10] - 1] = a[i];

count[(a[i] / place) % 10]--;

}

for (int i = 0; i < n; i++)

a[i] = output[i];

}

void radixsort(int a[], int n)

{

int max = getMax(a, n);

for (int place = 1; max / place > 0; place \*= 10)

countingSort(a, n, place);

}

int main()

{

int count = 0;

printf("Enter size of array : ");

scanf("%d", &count);

int a[count];

for (int i = 0; i < count; i++)

{

printf("Enter the %d element : " ,i+1);

scanf("%d", &a[i]);

}

radixsort(a, count);

printf("Displaying Sorted array in ascending order: \n");

for (int i = 0; i < count; ++i)

printf("%d ", a[i]);

printf("\n");

return 0;

}

