****

**KRISHNA ENGINEERING COLLEGE**

**Data Structure Lab**

**KCS – 301**

SUBMITTED BY:-

**ANKIT GUSAIN**

**1901610100021**

SUBMITTED TO:-

**Ms. RASHMI SHARMA**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Krishna Engineering College**  **Department of Computer Science and Engineering** | | | |  |
| **Name: ANKIT GUSAIN**  **Roll No.:1901610100021** | | |  |
|  |
| **Year/Semester:- 2nd/III** | | **Session:- 2020-21** |  |
| **Subject Name:- Data Structures** | | **Subject Code:- KCS-351** |  |
| **Faculty :-** | **Ms. Rashmi** | **Section/ Group: 2/B** |  |
|  |  |  |  |
| **S.No** | **Experiment /Program Name** | **Scheduled Date** | **Sign** |
|
| 1 | Traverse of array |  |  |
| 2 | Implementation of Linear search |  |  |
| 3 | Implementation of Binary search |  |  |
| 4 | Insertion sort |  |  |
| 5 | Selection sort |  |  |
| 6 | Bubble sort |  |  |

**Q11. WAP to implement Linear Search in C.**

#include <stdio.h>

#include <conio.h>

int main()

{

int a[6],i,key,found=1;

printf("enter only six elements in array\n");

for(i=0;i<6;i++)

{

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

}

printf("enter the element to be search\n");

scanf("%d",&key);

for(i=0;i<6;i++)

{

if(a[i]==key)

{

printf("the element is in %d position\n",i);

break;

}

}

return 0;

}

Output

enter only six elements in array

1

2

3

4

5

6

enter the element to be search

3

the element is in 2 position

**Q2. WAP to implement BINARY Search in C.**

#include <stdio.h>

int main()

{

int i, low, high, mid, n, key, array[100];

printf("Enter number of elements");

scanf("%d",&n);

printf("Enter %d integers", n);

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

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

printf("Enter value to find");

scanf("%d", &key);

low = 0;

high = n - 1;

mid = (low+high)/2;

while (low <= high) {

if(array[mid] < key)

low = mid + 1;

else if (array[mid] == key) {

printf("%d found at location %d.n", key, mid+1);

break;

}

else

high = mid - 1;

mid = (low + high)/2;

}

if(low > high)

printf("Not found! %d isn't present in the list.n", key);

return 0;

}

Output

Enter number of elements6

Enter 6 integers1

2

3

4

5

6

Enter value to find4

4 found at location 4.

**Q3. WAP to traverse array in C.**

#include <stdio.h>

int main()

{

int n =5;

int a[] = {3, 6, 9, 12, 15};

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

{

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

}

return 0;

}

Output

3 6 9 12 15

**Q4. WAP to implement insertion sort in C.**

#include <math.h>

#include <stdio.h>

void insertionSort(int arr[], int n)

{

int i, temp, j;

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

{

temp = arr[i];

j = i - 1;

while (j >= 0 && arr[j] > temp)

{

arr[j + 1] = arr[j];

j = j - 1;

}

arr[j + 1] = temp;

}

}

void printArray(int arr[], int n)

{

int i;

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

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

printf("\n");

}

int main()

{

int n;

int arr[20];

printf("enter the size of array");

scanf("%d", &n);

printf("enter the elements of array");

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

{

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

}

insertionSort(arr, n);

printArray(arr, n);

return 0;

}

**Output:**

enter the elements of array

41

31

21

11

11 21 31 41

**Q5. WAP to implement selection sort in C.**

#include <stdio.h>

void swap(int \*x, int \*y)

{

int temp = \*x;

\*x = \*y;

\*y = temp;

}

void selectionSort(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 printArray(int arr[], int size)

{

int i;

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

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

printf("\n");

}

int main()

{

int n;

int arr[20];

printf("enter the size of array");

scanf("%d", &n);

printf("enter the elements of array");

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

{

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

}

selectionSort(arr, n);

printf("Sorted array: \n");

printArray(arr, n);

return 0;

}

**Output:**

enter the elements of array

41

31

21

11

11 21 31 41

**Q4. WAP to implement bubble sort in C.**

#include <stdio.h>

void display(int a[10], int n)

{

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

{

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

}

}

int main()

{

int a[10], n, temp = 0;

printf(" BUBBLE SORTING ");

printf("\nEnter Array size :");

scanf("%d", &n);

printf(" Enter Array Elements:\n");

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

{

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

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

}

printf("\nUnsorted Array : ");

display(a, n);

for (int i = 0; i < n - 1; i++)

{

for (int j = 0; j < n - 1 - i; j++)

{

if (a[j] > a[j + 1])

{

temp = a[j + 1];

a[j + 1] = a[j];

a[j] = temp;

}

}

}

printf("\nSorted Array after bubble sort : ");

display(a, n);

return 0;

}

**Output:**

Enter Array size :4

Enter Array Elements:

a1:42

a2:23

a3:31

a4:11

Unsorted Array : 42 23 31 11

Sorted Array after bubble sort : 11 23 31 42