CODES OF ARRAY

* BASIC:

#include <iostream>

int main() {

std::cout << "Hello World!";

return 0;

}

* C++ Program to traverse the array:

#include <iostream>

using namespace std;

int main()

{

int arr[5]={10, 0, 20, 0, 30}; //creating and initializing array

//traversing array

for (int i = 0; i < 5; i++)

{

cout<<arr[i]<<"\n";

}

}

* C++ Program for sequential search in an array

#include<iostream>

using namespace std;

int main()

{

int arr[10],i,num,index;

cout<<"Enter the elemnts";

for(i=0;i<10;i++)

cin>>arr[i];

cout<<"Enter the number";

cin>>num;

for(i=0;i<10;i++)

{

if(arr[i]==num)

{

index=i;

break;

}

}

cout<<"Element found"<<index;

cout<<endl;

return 0;

}

* C++ Program to insert an element in the array based on its position

#include<iostream>

using namespace std;

int main()

{

int arr[50], i, elem, pos, tot;

cout<<"Enter the Size for Array: ";

cin>>tot;

cout<<"Enter "<<tot<<" Array Elements: ";

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

cin>>arr[i];

cout<<"\nEnter Element to Insert: ";

cin>>elem;

cout<<"At What Position ? ";

cin>>pos;

for(i=tot; i>=pos; i--)

arr[i] = arr[i-1];

arr[i] = elem;

tot++;

cout<<"\nThe New Array is:\n";

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

cout<<arr[i]<<" ";

cout<<endl;

return 0;

}

* C++ Program to delete an element in the array based on its position

#include <iostream>

using namespace std;

int main() {

int a[100], size, pos, i, count=0;

cout << "Enter the size of an array \n";

cin >> size;

cout << "Enter the value in an array \n";

// Take an input array

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

cin >> a[i];

}

//Input position where we delete an element

cout << "Enter the position \n";

cin >> pos;

//Shift element from i+1 to i

for(i = pos-1; i < size; i++)

{

arr[i] = arr[i+1];

}

// Reduce the size of an array

size--;

// Print an array after deleting an element

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

cout<<" "<<a[i];

}

return 0;

}

* C++ program to delete an element in the array

#include<iostream>

using namespace std;

int main()

{

int arr[100], tot, i, elem, j, found=0;

cout<<"Enter the Size: ";

cin>>tot;

cout<<"Enter "<<tot<<" Array Elements: ";

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

cin>>arr[i];

cout<<"\nEnter Element to Delete: ";

cin>>elem;

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

{

if(arr[i]==elem)

{

for(j=i; j<(tot-1); j++)

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

found=1;

i--;

tot--;

}

}

if(found==0)

cout<<"\nElement doesn't found in the Array!";

else

{

cout<<"\nElement Deleted Successfully!";

cout<<"\n\nThe New Array is:\n";

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

cout<<arr[i]<<" ";

}

cout<<endl;

return 0;

}

* c++ program to traverse the whole array and find the average

#include<iostream>

using namespace std;

double average(int array[], int size)

{

double sum= 0.0;

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

{

sum = sum+array[i];

}

return (sum/size);

}

int main()

{

int points[]={18,20,5,17,22};

cout << average (points, 5) << endl;

}

* Ques 1 a) Create a C++ program with array with datatypes as string, char and int in unsorted manner?

b) Traverse the whole array and print.

c) Sort in DESCENDING order.

#include<iostream>

using namespace std;

int main()

{

int a,b,c;

cout<<"Enter size for string array"<<endl;

cin>>a;

cout<<"Enter size for char array"<<endl;

cin>>b;

cout<<"Enter size for int array"<<endl;

cin>>c;

string arr1[a],temp1;

char arr2[b],temp2;

int arr3[c],temp3;

cout<<"Enter elements in string array"<<endl;

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

{

cin>>arr1[i];

}

cout<<"Enter elements in char array"<<endl;

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

{

cin>>arr2[i];

}

cout<<"Enter elements in int array"<<endl;

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

{

cin>>arr3[i];

}

cout<<"String Array"<<endl;

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

{

cout<<arr1[i]<<" ";

}

cout<<endl;

cout<<"Char Array"<<endl;

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

{

cout<<arr2[i]<<" ";

}

cout<<endl;

cout<<"Int Array"<<endl;

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

{

cout<<arr3[i]<<" ";

}

cout<<endl;

//sorting string array in descending order

for(int i=0;i<5;i++)

{

for(int j=i+1;j<5;j++)

{

if(arr1[i]<arr1[j])

{

temp1=arr1[i];

arr1[i]=arr1[j];

arr1[j]=temp1;

}

}

}

cout<<"String Array sorted in descending order"<<endl;

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

{

cout<<arr1[i]<<" ";

}

cout<<endl;

//sorting char array in descending order

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

{

for(int j=i+1;j<b;j++)

{

if(arr2[i]<arr2[j])

{

temp2=arr2[i];

arr2[i]=arr2[j];

arr2[j]=temp2;

}

}

}

cout<<"Char Array sorted in descending order"<<endl;

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

{

cout<<arr2[i]<<" ";

}

cout<<endl;

//sorting int array in descending order

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

{

for(int j=i+1;j<c;j++)

{

if(arr3[i]<arr3[j])

{

temp3=arr3[i];

arr3[i]=arr3[j];

arr3[j]=temp3;

}

}

}

cout<<"Int Array sorted in descending order"<<endl;

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

{

cout<<arr3[i]<<" ";

}

cout<<endl;

return 0;

}

* C++ Program Binary Search Iterative

#include <iostream>

using namespace std;

int binarySearch(int array[], int x, int low, int high) {

// Repeat until the pointers low and high meet each other

while (low <= high) {

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

if (array[mid] == x)

return mid;

if (array[mid] < x)

low = mid + 1;

else

high = mid - 1;

}

return -1;

}

int main(void) {

int array[] = {3, 4, 5, 6, 7, 8, 9};

int x = 4;

int n = sizeof(array) / sizeof(array[0]);

int result = binarySearch(array, x, 0, n - 1);

if (result == -1)

printf("Not found");

else

printf("Element is found at index %d", result);

}

* C++ Program Binary Search- Recursive

#include <iostream>

using namespace std;

int binarySearch(int array[], int x, int low, int high) {

if (high >= low) {

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

// If found at mid, then return it

if (array[mid] == x)

return mid;

// Search the left half

if (array[mid] > x)

return binarySearch(array, x, low, mid - 1);

// Search the right half

return binarySearch(array, x, mid + 1, high);

}

return -1;

}

int main(void)

{

int array[] = {3, 4, 5, 6, 7, 8, 9};

int x = 4;

int n = sizeof(array) / sizeof(array[0]);

int result = binarySearch(array, x, 0, n - 1);

if (result == -1)

printf("Not found");

else

printf("Element is found at index %d", result);

}

* SORTING TECHNIQUES
* Bubble Sort

#include <iostream>

using namespace std;

void swap(int \*xp, int \*yp)

{

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

// A function to implement bubble sort

void bubbleSort(int arr[], int n)

{

int i, j;

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

// Last i elements are already in place

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

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

swap(&arr[j], &arr[j+1]);

}

/ Function to print an array \*/

void printArray(int arr[], int size)

{

int i;

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

cout << arr[i] << " ";

cout << endl;

}

// Driver code

int main()

{

int arr[] = {64, 34, 25, 12, 22, 11, 90};

int n = sizeof(arr)/sizeof(arr[0]);

bubbleSort(arr, n);

cout<<"Sorted array: \n";

printArray(arr, n);

return 0;

}

* Insertion Sort

#include<iostream>

using namespace std;

int main()

{

int n,i;

cin>>n;

int arr[n];

{

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

cin>>arr[i];

}

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

{

int current=arr[i];

int j=i-1;

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

{

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

j--;

}

arr[j+1]=current;

}

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

{

cout<<arr[i]<<"";

}

cout<<endl;

}

* BUBBLE SORT SHORT
* **void** bubbleSort(**int** arr[], **int** n)
* {
* **int** i, j;
* **for** (i = 0; i < n - 1; i++)
* // Last i elements are already
* // in place
* **for** (j = 0; j < n - i - 1; j++)
* **if** (arr[j] > arr[j + 1])
* swap(arr[j], arr[j + 1]);
* }
* Selection Sort

#include <iostream>

using namespace std;

void swap(int xp, int \*yp)

{

int temp = \*xp;

\*xp = \*yp;

\*yp = temp;

}

void selectionSort(int arr[], int n)

{

int i, j, min\_idx;

// One by one move boundary of unsorted subarray

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

{

// Find the minimum element in unsorted array

min\_idx = i;

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

if (arr[j] < arr[min\_idx])

min\_idx = j;

// Swap the found minimum element with the first element

swap(&arr[min\_idx], &arr[i]);

}

}

/ Function to print an array \*/

void printArray(int arr[], int size)

{

int i;

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

cout << arr[i] << " ";

cout << endl;

}

// Driver program to test above functions

int main()

{

int arr[] = {64, 25, 12, 22, 11};

int n = sizeof(arr)/sizeof(arr[0]);

selectionSort(arr, n);

cout << "Sorted array: \n";

printArray(arr, n);

return 0;

}

LINK FOR 3 SORTS VIDEOS

Bubble Sort: https://www.youtube.com/watch?v=nmhjrI-aW5o

Insertion Sort: https://www.youtube.com/watch?v=OGzPmgsI-pQ

Selection Sort: <https://www.youtube.com/watch?v=xWBP4lzkoyM>