Arrays (class 1)

10-02-2023

Date...................

* write the significance of 0 Based indexing in array?

why array indexing starts from 0?

	0	1	2	3	4
arr →	10	20	30	40	50

name of the array stores the 0th index address.

integer type array whose name is 'arr'.

Indexing of array starts from 0 because,

arr + 0 x 4

address of first value

index of first value

size of the type of first value, i.e., 4 Bytes

this is how compiler calculates the address of first value in the address.

Date.....................

Ques Linearly search an element in an array.

Linear search is a sequential searching algorithm where we start from one end & check every element of the list until the desired element is found. If the element not found then we return "not found".

Approach: we've given an array & the size of the array, i.e., n.
First we take input from the user that which element the user want to search.
Then we run the loop from 0th index to n-1 index.
At each iteration we compare the array element with the user's given search element.
If the element matched then we change the value of flag variable to 1.
At the end we check, if the flag = 1 then print "element found" else print "element not found".

code :-

```
#include<iostream>
using namespace std;
int main()
{
    int arr[] = {15, 58, 22, 10, 6, 1};
    int n = 6;
    int i, search, flag = 0;
    cout << "\n enter element: ";
    cin >> search;
    for(i=0; i<n; i++)
    {
        if (arr[i] == search)
        {
            flag = 1;
            break;
        }
        else
        {
            flag = 0;
        }
    }
    if (flag ==1)
    {
      cout << " Found ";
    }
    else
    {
      cout << "Not found ";
    }
    return 0;
}
```

Date.................

Ques. count 0's & 1's in an array.

In this problem, we've given an array & we have to count the number of times 0 present in array & number of times 1 present in array.

Approach: we've given an array full of 0's & 1's, & also the size of the array is given.
Firslty, we take two variables, i.e., numZero & numOne to count, & initialize them to zero.
Then we run a loop from 0th index to n-1 index.
At each iteration we check, if the element is 0, then increment numZero to 1, if the element is 1, then increment numOne to 1.
At the end, print both the variables.

code:-

```
#include<iostream>
using namespace std;
int main()
{
   int arr[] = {0,1,1,0,0,1,1,0};
   int n = 8;
   int numZero = 0, numOne = 0;
   for(int i = 0; i < n; i++)
   {
      if(arr[i] == 0)
      {
         numZero++;
      }
      if(arr[i] == 1)
      {
         numOne++;
      }
   }
   cout << "\n Number of 0's : " <<
                               numZero;
   cout << "\n Number of 1's are : "
                         << numOne;
   return 0;
}
```

Date..................

Ques. Find the maximum & minimum number in an array.

Code:-

```
#include<iostream>
using namespace std;
int main()
{
    int arr[] = {5, 1, 7, 6, 9, 8, 12 };
    int n = 7;
    int max = INT_MIN;
    for(int i = 0; i < n; i++)
    {
        if(arr[i] > max)
        {
            max = arr[i];
        }
    }
    cout << "\nMax no. is " << max;
    return 0;
}
```

Code:-

```
#include<iostream>
using namespace std;
int main()
{
   int arr[] = {5, 7, 3, 9, 8, 6};
   int n = 6;
   int min = INT_MAX;
   for(int i = 0; i < n; i++)
   {
      if(arr[i] < min)
      {
         min = arr[i];
      }
   }
   cout << "Min no. is :" << min;
   return 0;
}
```

Que) Extreme prints in an array.

In this problem, we print the extreme values in the array.
Means,

0	1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9	10

Extreme prints means, firstly the left most element is printed then the right most element is printed. Then left is incremented by one & right is decremented by one., & so on.

Approach: In this type of problem we follow "Two Pointer Approach", set i to 0th index & j to n-1 index. Then print both values, then increment i & decrement j, & follow the same thing again.

when to step out from the loop?
when i >= j, then we stop the iteration.

code :- (even case)

```
#include<iostream>
using namespace std;
int main()
{
    int arr[] = {1, 2, 3, 4, 5, 6, 7, 8 };
    int n = 8;
    cout << "\nExtreme print :";
    for(int i = 0, j = n-1; i < j;
                            i++, j--)
    {
        cout << arr[i] << " " <<
             arr[j] << " ";
    }
    return 0;
}
```

code :- (odd case)

```
int main()
{
    int arr[] = {1, 2, 3, 4, 5, 6, 7 };
    int n = 7;
    int i = 0, j = n-1;
    cout << "extreme prints :";
    while(i <= j)
    {
        if(i == j){
            cout << arr[i] << " ";
        }
        else{
```

```
        cout << arr[i] << " " <<
             arr[j] << " ";
      }
      i++;
      j--;
    }
    return 0;
}
```

Date..................

Ques. Reverse an array. (swapping)

code :-

```
#include<iostream>
using namespace std;
int main ()
{
    int arr[] = {10, 20, 30, 40, 50 };
    int n = 5;
    cout << "\nBefore Swapping :";
    for(int i = 0; i < n; i++)
    {
        cout << arr[i] << "  ";
    }
    int i = 0, j = n-1;
    while (i <= j)
    {
        swap (arr[i], arr[j]);
        i++;
        j--;
    }
    cout << "\nAfter swapping :";
    for(int i = 0; i < n; i++)
    {
        cout << arr[i] << "  ";
    }
    return 0;
}
```