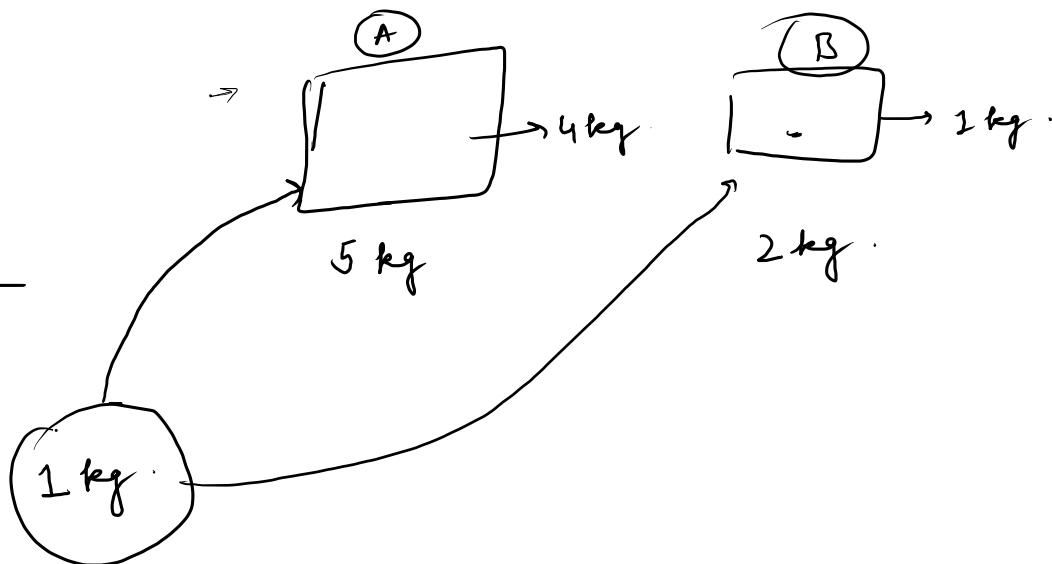


①

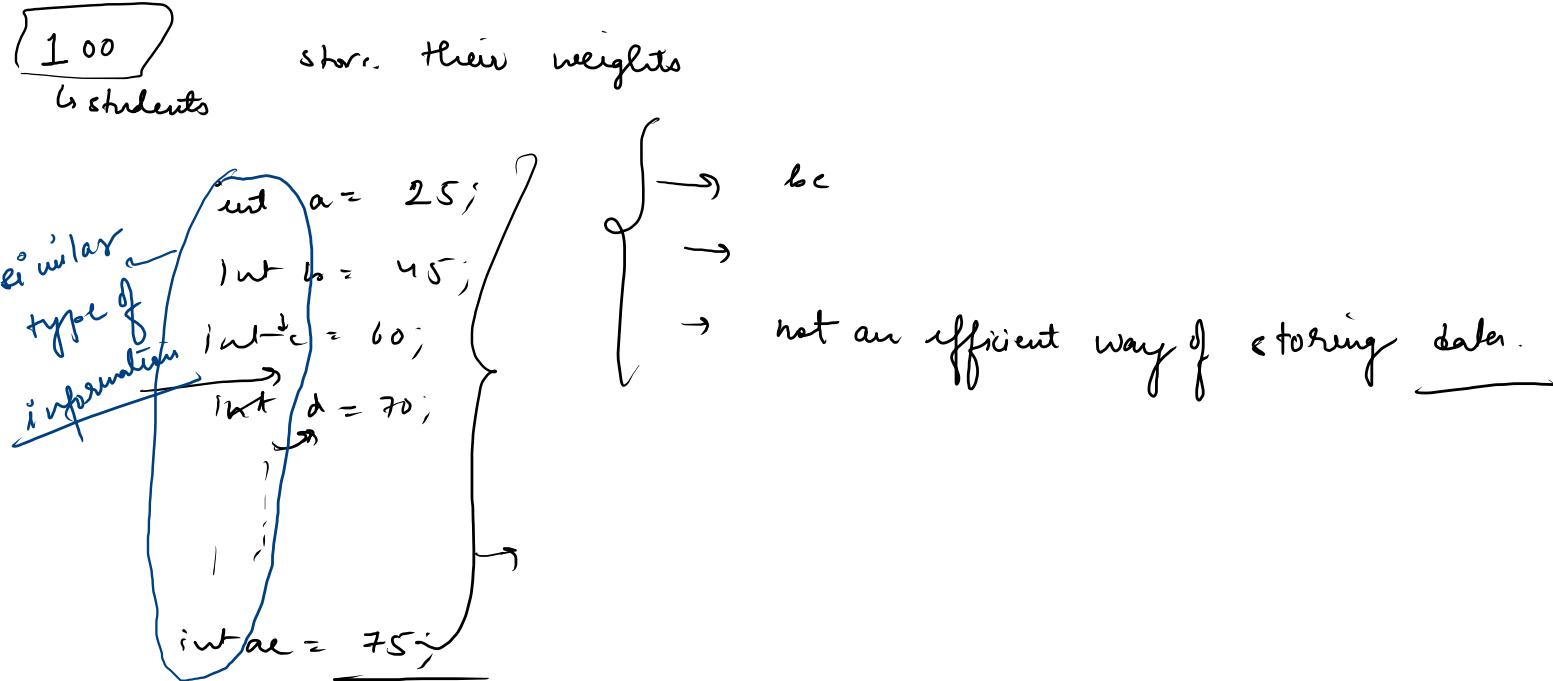
Time and Space complexity

2

Arrays



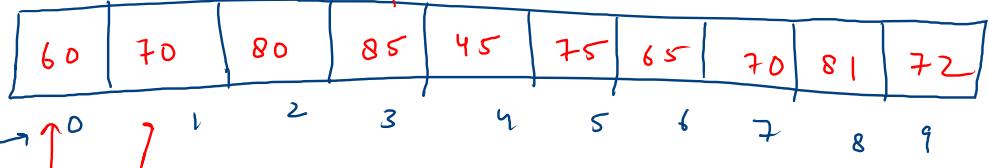
Array :-



Why array

10 students

arr →

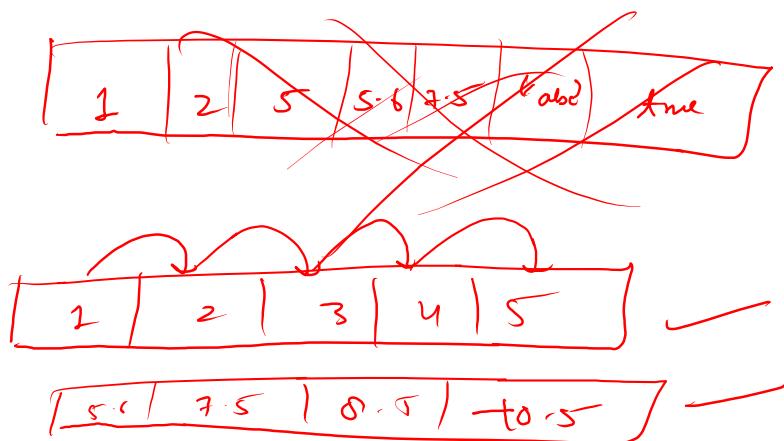


wt
int
1 wt
r

similar

Array:- It is a data structure which is used to store similar type of information in a continuous manner.

- ↳ int
- ↳ float
- ↳ boolean
- ↳ character
- ↳ string



Syntax of Array

→ Declare and initialization :

~~int~~ datatype array-name [] = new ~~datatype~~ [size];

datatype [] array-name = new datatype []

Ex:- Write an array syntax for storing 5 elements? (int)

int arr [] = new int [5];

or

int [] arr = new int [5];

int arr [];

arr = new int [5];

Declaration

Initialization

declaration and initialization

Memory Management of Array

`int arr[5];` → Declaration
`arr = new int[5];` → Initialization

$$\begin{aligned} 1 \text{ int} &\rightarrow 4 \text{ bytes} \\ 5 \text{ int} &\rightarrow 4 \times 5 = 20 \text{ bytes} \end{aligned}$$

→ So 20 bytes is allocated in heap.

→ length of array

`arr.length`

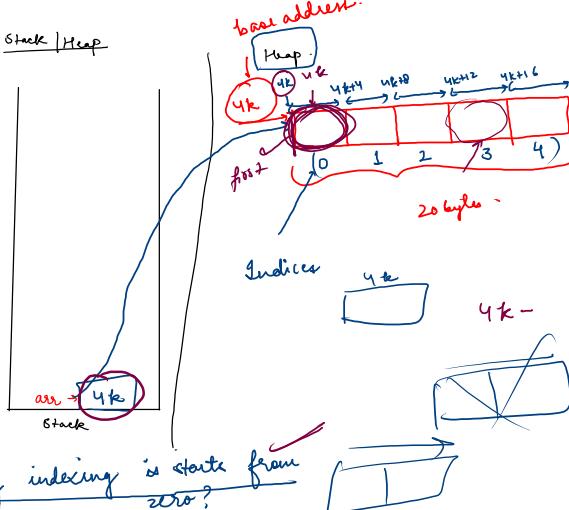
`array-name.length`

⇒ 5 Q why indexing starts from zero?

→ How far you are from base address

5 minutes

Stack / Heap



$$\text{Index} = \frac{\text{Element address} - \text{base address}}{\text{datatype size}}$$

$$= \frac{4k + 12 - 4k}{4} = \frac{0}{4} = 0$$

$$= \frac{4k + 12 - 4k}{4} = \frac{12}{4} = 3$$

Print the array elements linewise (1 july)

Problem

Submissions

Leaderboard

Discussions

Take n as an integer input. Declare an array of size n that stores value of int data-type.

Then take n integer inputs and store them in the array one by one.

And print each integer in each line.

Output Format

$$n = 5$$

Print the array element in different line.

Sample Input 0

```
5  
1  
2  
3  
4  
5
```

Sample Output 0

```
1  
2  
3  
4  
5
```

int arr[] = new int [n];

array's use I/P.

```

import java.io.*;
import java.util.*;

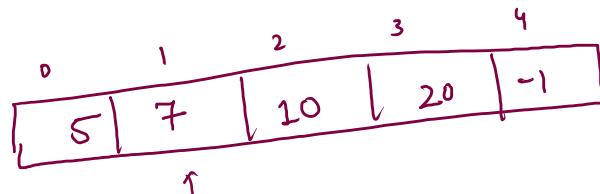
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT */
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt(); → n=5

        int arr [] = new int[n]; →

        // User Input for array
        for(int i = 0;i<n;i++){
            arr[i] = scn.nextInt();
        }
    }
}

```



$i=0 \leq 5 \text{ (T)}$ $\text{arr}[0] = 5$
 $i=1 \leq 5 \text{ (T)}$ $\text{arr}[1] = 7$
 $i=2 \leq 5 \text{ (T)}$ $\text{arr}[2] = 10$
 $i=3 \leq 5 \text{ (T)}$ $\text{arr}[3] = 20$
 $i=4 \leq 5 \text{ (T)}$ $\text{arr}[4] = -1$
 $i=5 \leq 5 \text{ (F)} \times$

Without function

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Pr
Scanner scn = new Scanner(System.in);
int n = scn.nextInt();

int arr [] = new int[n];

//      User Input for array
for(int i = 0;i<n;i++){
    arr[i] = scn.nextInt();
}

//      Printing elements of array
for(int i=0;i<n;i++){
    System.out.println(arr[i]);
}
}
}
```

With function

```
public class Solution {

    public static void printArrayElements(int arr[]){
        // int n = arr.length;
        for(int i=0;i<arr.length;i++){
            System.out.println(arr[i]);
        }
    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Pr
Scanner scn = new Scanner(System.in);
int n = scn.nextInt();

int arr [] = new int[n];

//      User Input for array
for(int i = 0;i<n;i++){
    arr[i] = scn.nextInt();
}

printArrayElements(arr);
}
}
```

Find Element In An Array (1 july) / Linear search

Problem

Submissions

Leaderboard

Discussions

1. You are given a number n , representing the size of array a .
2. You are given n distinct numbers, representing elements of array a .
3. You are given another number d .
4. You are required to check if d number exists in the array a and at what index (0 based). If found print the index, otherwise print -1.

Sample Input 0

6
15
30
40
4
11
9
48

$\rightarrow T.C \Rightarrow O(n)$
 $\rightarrow S.C \Rightarrow O(1)$

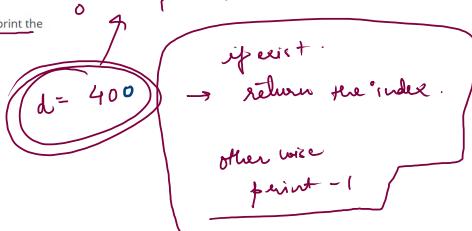
Sample Output 0

2

i = 1
2
3
4
5

$n = 6 \rightarrow$ size of array.
requirement of question

arr	[15		30		40		4		11		9]
		0		1		2		3		4		5	



```
for(int i=0; i<n; i++) {  
    if (arr[i] == d)  
        return i;  
}  
return -1;
```

→ main fn.

$i = 0 < 6 (T)$
 $arr[0] = 15 == 400 X$

$i = 1 < 6 (T)$
 $arr[1] = 30 == 400 X$

$i = 2 < 6 (T)$
 $arr[2] = 40 == 400 \cancel{X}$

$i = 3 < 6 (T)$
 $arr[3] = 4 == 400 X$

$i = 4 < 6 (T)$
 $arr[4] = 11 == 400 X$

$i = 5 < 6 (T)$
 $arr[5] = 9 == 400 X$

$i = 6 < 6 (F)$

→ return -1

```
import java.io.*;
import java.util.*;

public class Solution {
    public static int linearSearch(int arr[], int d){
        for(int i =0;i<arr.length;i++){
            if(arr[i] == d){
                return i;
            }
        }
        return -1;
    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT */
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int arr [] = new int[n];

        for(int i =0;i<n;i++){
            arr[i] = scn.nextInt();
        }

        int d = scn.nextInt();

        System.out.println(linearSearch(arr,d));
    }
}
```

$d = 2 \rightarrow -1$

0 1 2 3 4
| 5 | 4 | 3 | 2 | 1

-1
-1
-1
3

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from
        STDIN. Print output to STDOUT. Your class
        should be named Solution. */
        Scanner scn=new Scanner(System.in);
        int n=scn.nextInt();
        int d=scn.nextInt();
        int arr[]= new int[n];
        for(int i=0;i<n;i++){
            arr[i]=scn.nextInt();
        }
        for(int i=0;i<arr.length;i++){
            if (d==arr[i]){
                System.out.print(i);
            } else{
                System.out.println("-1");
            }
        }
    }
}
```

$i = 0 \leftarrow 5 \text{ (T)}$
 $2 = = 5 \times$
 $i = 1 \leftarrow 5 \text{ (T)}$
 $2 = = 4 \times$
 $i = 2 \leftarrow 5 \text{ (T)}$
 $2 = = 3 \times$
 $i = 3 \leftarrow 5 \text{ (T)}$
 $2 = = 2 \times$

Min_Max_Array (1 july)

Problem

Submissions

Leaderboard

Discussions

1. You are given a number n, representing the count of elements.

2. You are given n numbers.

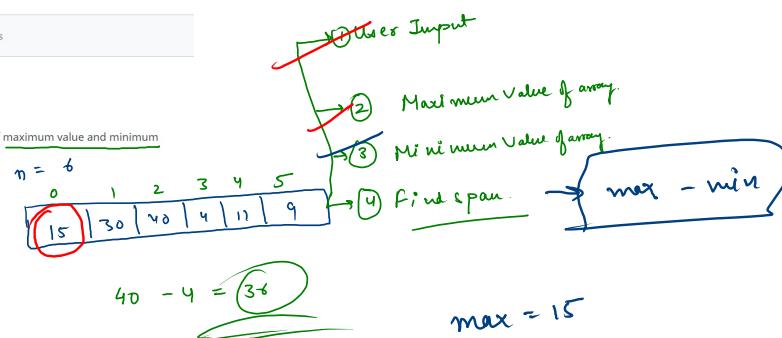
3. You are required to find the span of input. Span is defined as difference of maximum value and minimum value.

Sample Input 0

```
6  
15  
30  
40  
4  
11  
9
```

Sample Output 0

36



```
int max = arr[0]; min = arr[0] i=1 < 6(T)  
for (int i=1; i < n; i++) arr[1] = 30 > 15(T)  
{ if (arr[i] > max) max = arr[i]; i=2 < 6(T)  
    max = 30  
    arr[2] = 40 > 30(T)  
    max = 40  
    arr[3] = 4 > 40 X  
    return max;  
}
```

$i = 4 < 6(T)$
 $arr[4] = 11 > 40X$
 $i = 5 < 6(T)$
 $arr[5] = 9 > 40X$
 $i = 6 < 6(T)$

```

public static int maxOfArray(int arr[]){
    int max = arr[0];
    for(int i=1;i<arr.length;i++){
        if(arr[i] > max){
            max = arr[i];
        }
    }
    return max;
}

public static int minOfArray(int arr[]){
    int min = arr[0];
    for(int i=1;i<arr.length;i++){
        if(arr[i] < min){
            min = arr[i];
        }
    }
    return min;
}

public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int arr[] = new int[n];

    for(int i=0;i<n;i++){
        arr[i] = scn.nextInt();
    }

    int maxElement = maxOfArray(arr);
    int minElement = minOfArray(arr);
    int span = maxElement - minElement;

    System.out.println(span);
}

```

~~Time Complexity~~

$T.C = O(n)$

$S.C = O(1)$

$O(n)$

$O(n)$

$O(n)$

$O(1)$

```
import java.io.*;
import java.util.*;

public class Solution {
    public static int spanOfArray(int arr[]){
        int min = arr[0];
        int max = arr[0];
        for(int i=1;i<arr.length;i++){
            if(arr[i] <min){
                min = arr[i];
            }
            if(arr[i]>max){
                max = arr[i];
            }
        }
        return max-min;
    }

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT */
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int arr[] = new int[n];

        for(int i=0;i<n;i++){
            arr[i] = scn.nextInt();
        }

        System.out.println(spanOfArray(arr));
    }
}
```

```
public static int maxOfArray(int arr[]){
    int max = arr[0];
    for(int i=1;i<arr.length;i++){
        if(arr[i] > max){
            max = arr[i];
        }
    }
    return max;
}

public static int minOfArray(int arr[]){
    int min = arr[0];
    for(int i=1;i<arr.length;i++){
        if(arr[i] <min){
            min = arr[i];
        }
    }
    return min;
}

public static void main(String[] args) {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int arr[] = new int[n];

    for(int i=0;i<n;i++){
        arr[i] = scn.nextInt();
    }

    int maxElement = maxOfArray(arr);
    int minElement = minOfArray(arr);
    int span = maxElement - minElement;

    System.out.println(span);
}
```