1) Given an array nums containing n distinct numbers in the range [1, n], return the only number in the range that is missing from the array.

Sample Test Case:

Enter the array elements separated by space: 1 3 4 5

2

Program:

using System;

class Task1{

static void Main(string[] args){

Console.WriteLine("Enter the array elements separated by space: ");

string[] input = Console.ReadLine().Split(' ');

int[] nums = new int[input.Length];

for(int i = 0; i<input.Length; i++){

nums[i] = int.Parse(input[i]);

}

int n = nums.Length+1;

int sum = (n\*(n+1))/2;

int Arrsum=0;

foreach(int num in nums){

Arrsum+=num;

}

int miss = sum - Arrsum;

Console.WriteLine(miss);

}

}

Output:

C:\Users\aryas\OneDrive\Desktop\cs\Day 5>Task1

Enter the array elements separated by space:

1 3 4 5 6

2

2) Given a non-negative integer x, return the square root of x rounded down to the nearest integer. The returned integer should be non-negative as well.

You must not use any built-in exponent function or operator.

Sample Test Case:

Input :

16

Output:

Square root: 4

Program:

Output:

3) Given an integer n, return true if it is a power of two. Otherwise, return false.

An integer n is a power of two, if there exists an integer x such that n == 2x.

Sample Test Case:

16

True

Explanation:

2^4 = 16

Program:

using System;

class Task3{

static void Main(string[] args){

int flag=1;

int num = int.Parse(Console.ReadLine());

while(flag<num){

flag\*=2;

if(flag == num){

Console.WriteLine("True");

}

}

if(flag!=num){

Console.WriteLine("False");

}

}

}

Output:

C:\Users\aryas\OneDrive\Desktop\cs\Day 5>Task3

16

True

C:\Users\aryas\OneDrive\Desktop\cs\Day 5>Task3

35

False

4)

Given a non-empty array of integers nums, every element appears twice except for one. Find that single one. You must implement a solution with a linear runtime complexity and use only constant extra space.

Sample Test Case:

Enter the array elements separated by space: 100 34 90 100 90

Single number: 34

Program:

using System;

class Task3{

static void Main(string[] args){

Console.WriteLine("Enter the array elements separated by space:");

string[] input = Console.ReadLine().Split(' ');

int[] numbers = new int[input.Length];

for (int i = 0; i < input.Length; i++){

numbers[i] = int.Parse(input[i]);

}

int singleNumber = 0;

for (int i = 0; i < numbers.Length; i++){

int count = 0;

for (int j = 0; j < numbers.Length; j++){

if (numbers[i] == numbers[j]){

count++;

}

}

if (count == 1){

singleNumber = numbers[i];

break;

}

}

Console.WriteLine("Single number: " + singleNumber);

}

}

Output:

C:\Users\aryas\OneDrive\Desktop\cs\Day 5>Task4

Enter the array elements separated by space:

100 29 29 100 2

Single number: 2

5) numbers such that they add up to the target.

You may assume that each input would have exactly one solution, and you may not use the same element twice.

You can return the answer in any order.

Sample Test Case:

Enter the number of elements in the array: 6

Enter the array elements separated by space:

1 2 3 4 5 6

target sum: 3

Indices of the two numbers: 0, 1

Program:

using System;

class Program{

static void Main(){

Console.WriteLine("Enter the array elements separated by space:");

string[] input = Console.ReadLine().Split(' ');

int[] nums = new int[input.Length];

for (int i = 0; i < input.Length; i++){

nums[i] = int.Parse(input[i]);

}

Console.Write("Target sum: ");

int target = int.Parse(Console.ReadLine());

int[] result = new int[2];

for (int i = 0; i < nums.Length; i++){

for (int j = i + 1; j < nums.Length; j++){

if (nums[i] + nums[j] == target){

result[0] = i;

result[1] = j;

break;

}

}

if (result[0] != 0 || result[1] != 0) break;

}

Console.WriteLine("Indices of the two numbers: {0}, {1}", result[0], result[1]);

}

}

Output:

C:\Users\aryas\OneDrive\Desktop\cs\Day 5>Task5

Enter the array elements separated by space:

1 2 3 4 5 6 7

Target sum: 9

Indices of the two numbers: 1, 6

6) Given an integer array A of size N. You have to pick exactly B elements from either the left or right end of the array A to get the maximum sum.

Find and return this maximum possible sum.

NOTE: Suppose B = 4 and array A contains 10 elements then You can pick the first four elements or can pick the last four elements or can pick 1 from the front and 3 from the back etc. You need to return the maximum possible sum of elements you can pick.

Sample Test Case:

Enter the array elements separated by space: 2 3 4 5 6 8

4

23

Explanation:

Here the sum of 4 elements of the left end is 14

The sum of the 4 elements of the right end is 23

And the sum of 2 elements of the left end and 2 elements of the right end = 19

Similarly make such combinations with 4 elements and find out the sum,

Therefore the highest sum is 23

Program:

using System;

class Task6{

static void Main(string[] args){

Console.WriteLine("Enter the array elements separated WinSizey space:");

string[] input = Console.ReadLine().Split(' ');

int[] arr = new int[input.Length];

for (int i = 0; i < input.Length; i++){

arr[i] = int.Parse(input[i]);

}

Console.WriteLine("Enter Window Size:");

int WinSize = int.Parse(Console.ReadLine());

int N = arr.Length;

int leftSum = 0;

for (int i = 0; i < WinSize; i++){

leftSum += arr[i];

}

int rightSum = 0;

for (int i = N - 1; i >= N - WinSize; i--){

rightSum += arr[i];

}

int maxSum = Math.Max(leftSum, rightSum);

Console.WriteLine("Max sum: " + maxSum);

}

}

Output:

C:\Users\aryas\OneDrive\Desktop\cs\Day 5>Task6

Enter the array elements separatead WinSizey space:

3 4 5 6 7 8

Enter Window Size:

2

Max sum: 15

7) find the duplicate values from th given array

Program:

using System;

class Task7{

static void Main(){

Console.WriteLine("Enter the array elements separated with spaces");

string[] input = Console.ReadLine().Split(' ');

int[] arr = new int[input.Length];

Console.WriteLine("Duplicate values in the array:");

for (int i = 0; i < input.Length; i++)

{

for (int j = i + 1; j < input.Length; j++)

{

if (input[i] == input[j])

{

Console.WriteLine(input[i]);

}

}

}

}

}

Output:

C:\Users\aryas\OneDrive\Desktop\cs\Day 5>Task7

Enter the array elements separated with spaces

7 6 6 5 5 3 3

Duplicate values in the array:

6

5

3

8) remove duplicates from an array

9) Given an integer array nums, return true if any value appears at least twice in the array, and return false if every element is distinct.

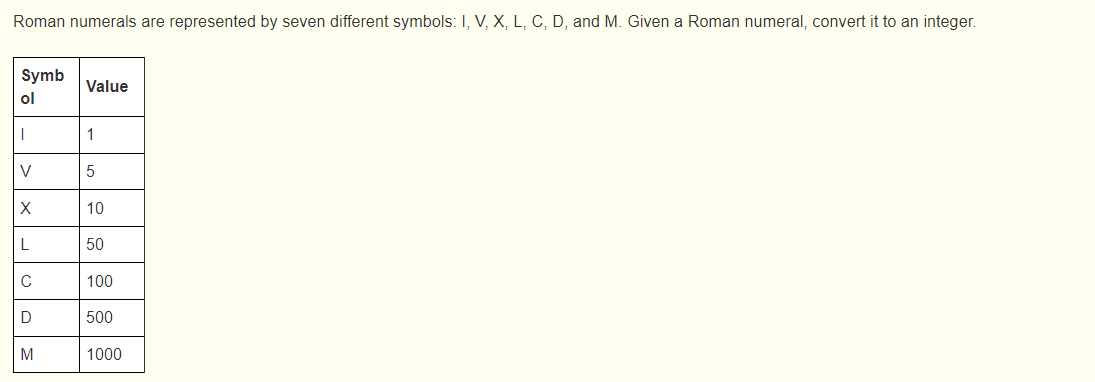
Sample Test Case:

22 33 44 55 22

true

Explanation:

Here in the given integer array 22 appeared twice in the array so the result is true.

10)

Program:

Output: