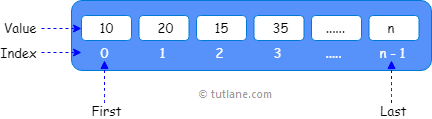
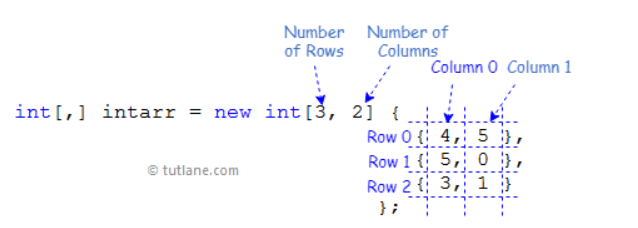
Array Exercise

Array Dimensional definition:

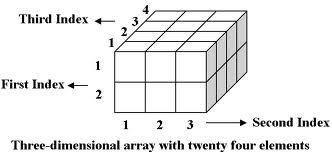
One Dimensional Array :



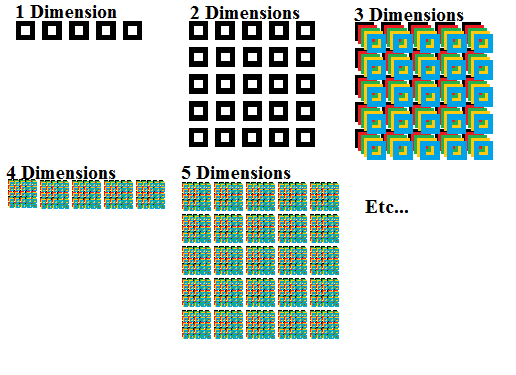
Two Dimensional Array :



Three Dimensional Array :



Int[i][j][k]=Int[2,4,3]



1. Receive one dimensional array from user

namespace excersice

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter the Length of your Array : ");

int arr\_length = Convert.ToInt32(Console.ReadLine());

int[] my\_arr = new int[arr\_length];

for (var i = 0; i < arr\_length; i++)

{

Console.WriteLine("please enter your value :");

my\_arr[i] = Convert.ToInt32(Console.ReadLine());

}

foreach(int j in my\_arr)

{

Console.WriteLine("My\_arr[{0}]", j);

}

}

}

}

1. Receive two dimensional array from user

using System;

namespace exercise

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter the number of your Rows : ");

int row = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the number of your columns : ");

int column = Convert.ToInt32(Console.ReadLine());

int[,] my\_arr = new int[row , column];

for (var i = 0; i < row; i++)

{

for (var j = 0; j < column; j++)

{

Console.WriteLine("please enter your row {0} values :" , i);

my\_arr[i , j] = Convert.ToInt32(Console.ReadLine());

}

}

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

{

for(int j = 0; j < column; j++)

{

Console.WriteLine("the row [{0}] and column [{1}] element : {2}" , i , j , my\_arr[i , j]);

}

}

}

}

}

Console.WriteLine("please enter your row {0} values :" , i);

my\_arr[i , j] = Convert.ToInt32(Console.ReadLine());

Receive three dimensional array from user

using System;

namespace excersice

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter the number of your Rows : ");

int row = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the number of your columns : ");

int column = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the number of your dimention : ");

int d = Convert.ToInt32(Console.ReadLine());

int[,,] my\_arr = new int[row , column , d];

for (var i = 0; i < row; i++)

{

for (var j = 0; j < column; j++)

{

for (int z = 0; z < d; z++)

{

Console.WriteLine("please enter your row {0} and column {1} values :", i ,j);

my\_arr[i, j , z] = Convert.ToInt32(Console.ReadLine());

}

}

}

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

{

for(int j = 0; j < column; j++)

{

for (int z = 0; z < d; z++)

{

Console.WriteLine("the row [{0}] and column [{1}] and [{2}] element : {3}", i, j, z, my\_arr[i, j, z]);

}

}

}

Jagged Array

using System;

namespace Myapplication

{

class program

{

static void Main(string[] args)

{

int[][] myjarr = new int[2][];

myjarr[0] = new int[2] { 4, 6 };

myjarr[1] = new int[] { 4, 9, 3 };

int[][,] myjarr2 = new int[2][,];

myjarr2[0] = new int[,] { { 4, 9 }, { 3, 2 } };

myjarr2[1] = new int[,] { { 3, 4 }, { 4, 1 } };

int[][,,] myjarr3 = new int[2][,,];

myjarr3[0] = new int[2, 2, 3] { { { 5, 9, 1 }, { 4, 2, 3 } }, { { 8, 5, 1 }, { 8, 6, 3 } } };

myjarr3[1] = new int[2, 2, 1] { { { 4 }, { 5 } }, { { 8 }, { 9 } } };

}

}

}

Every double {} represent new row!

Jagged Array access Elements

One dimension

using System;

namespace Myapplicaion

{

class program

{

static void Main(string[] args)

{

int[][] myjarr = new int[2][];

myjarr[0] = new int[] { 4, 6, 9 };

myjarr[1] = new int[] { 5, 9, 3 };

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

{

Console.WriteLine("[{0}] : ", i);

for (int j = 0; j < myjarr[i].Length; j++)

{

Console.WriteLine("{0}" , myjarr[i][j]);

}

}

}

}

}

Jagged Array access Elements

namespace jaggedarray

{

class Program

{

static void Main(string[] args)

{

int[][,] myjarr2 = new int[2][,];

myjarr2[0] = new int[,] { { 4, 9 }, { 3, 2 } };

myjarr2[1] = new int[,] { { 3, 4 }, { 4, 1 } };

for (int i = 0; i < myjarr2.Length; i++)

{

int x = 0;

// GetLength method takes integer x which

// specifies the dimension of the array

for (int j = 0; j < myjarr2[i].GetLength(x); j++)

{

// Rank is used to determine the total

// dimensions of an array

for (int k = 0; k < myjarr2[j].Rank; k++)

Console.Write("Jagged\_Array[" + i + "][" + j + ", " + k + "]: "

+ myjarr2[i][j, k] + " ");

Console.WriteLine();

}

x++;

Console.WriteLine();

}

}

}

}

Jagged Array access Elements

using System;

namespace jaggedarray

{

class Program

{

static void Main(string[] args)

{

int[][,] myjarr2 = new int[2][,];

myjarr2[0] = new int[,] { { 4, 9 }, { 3, 2 } };

myjarr2[1] = new int[,] { { 3, 4 }, { 4, 1 } };

for (int i = 0; i < myjarr2.Length; i++)

{

int x = 0;

// GetLength method takes integer x which

// specifies the dimension of the array

for (int j = 0; j < myjarr2[i].GetLength(x); j++)

{

// Rank is used to determine the total

// dimensions of an array

for (int k = 0; k < myjarr2[j].Rank; k++)

Console.Write( "myjarr item row {0} column {1} dimensiona {2} = {3} " , i , j , k , myjarr2[i][j, k] );

Console.WriteLine();

}

x++;

Console.WriteLine();

}

}

}

}

**Array Sort**

Defining a method which except an array and sort it

using System;

namespace sortarray

{

class sort

{

public void Mymethod(int[] arr)

{

Array.Sort(arr);

foreach(int i in arr)

{

Console.WriteLine(i);

}

}

}

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Please enter the Length of array : ");

int arrlength = Convert.ToInt32(Console.ReadLine());

int[] myarr = new int[arrlength];

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

{

Console.WriteLine("enter array element :");

myarr[i] = Convert.ToInt32(Console.ReadLine());

}

sort obj = new sort();

obj.Mymethod(myarr);

}

}

}

**An array with an odd number of elements is said to be centered if all elements (except the middle one) are strictly greater than the value of the middle element. Note that only arrays with an odd number of elements have a middle element. Write a function that accepts an integer array and returns 1 if it is a centered array, otherwise it returns 0**

using System;

namespace Myapplication

{

class Array

{

public int Mymethod(int[] arr)

{

int mid = (arr.Length - 1) / 2;

int midnum = arr[mid];

if (arr == null || arr.Length % 2 == 0)

{

return 0;

}

else

{

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

{

if (i != mid && arr[i] <= midnum)

{

return 0;

}

}

}

return 1;

}

}

class program

{

static void Main()

{

Console.WriteLine("Please Enter Your array length :");

int len = Convert.ToInt32(Console.ReadLine());

int[] myarr = new int[len];

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

{

Console.WriteLine("enter your Array Item :");

myarr[i] = Convert.ToInt32(Console.ReadLine());

}

Array obj = new Array();

Console.WriteLine(obj.Mymethod(myarr));

}

}

}

1. **Write a function that takes an array of integers as an argument and returns a value based on the sums of the even and odd numbers in the array.**Let X = the sum of the odd numbers in the array and let Y = the sum of the even numbers. The function should return X – Y

using System;

namespace Myapplication

{

class Array

{

public int Mymethod(int[] arr)

{

int sumodd = 0;

int sumeven = 0;

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

{

if(arr[i] % 2 == 0)

{

sumeven += arr[i];

}

else

{

sumodd += arr[i];

}

}

return sumodd - sumeven;

}

}

class program

{

static void Main()

{

Console.WriteLine("Please Enter Your array length :");

int len = Convert.ToInt32(Console.ReadLine());

int[] myarr = new int[len];

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

{

Console.WriteLine("enter your Array Item :");

myarr[i] = Convert.ToInt32(Console.ReadLine());

}

Array obj = new Array();

Console.WriteLine(obj.Mymethod(myarr));

}

}

}

1. **Write a function that accepts a character array, a zero-based start position and a length. It should return a character array containing containing *length*characters starting with the *start*character of the input array.** The function should do error checking on the start position and the length and return null if the either value is not legal.  
   The function signature is:  
   **char[ ] f(char[ ] a, int start, int len)**

using System;

namespace Myapplication

{

class Char

{

public void Mymethod(char[] arr , int sp , int larr)

{

int sub = arr.Length - sp;

if(larr > arr.Length || larr > sub || larr == 0)

{

Console.WriteLine("null");

}

else

{

for(int i = sp; i <= larr; i++ )

{

Console.WriteLine("myarr [{0}] = {1}" , i , arr[i]);

}

}

}

}

class program

{

static void Main()

{

Console.WriteLine("Please Enter Your array length :");

int len = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Please Enter your array start position :");

int startp = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Please Enter your array lengthcharacters :");

int arrlen = Convert.ToInt32(Console.ReadLine());

char[] myarr = new char[len];

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

{

Console.WriteLine("enter your Array Item :");

myarr[i] = Convert.ToChar(Console.ReadLine());

}

Char obj = new Char();

obj.Mymethod(myarr , startp , arrlen );

}

}

}

1. **Write a function to reverse an integer using numeric operators and without using any arrays or other data structures.**  
   The signature of the function is:  
   **int f(int n)**

using System;

namespace Myapplication

{

class Reverse

{

public int Mymethod(int num)

{

int sign = 1;

int reverse = 0;

if (num == 0) { return 0; }

if(num < 0)

{

sign = -1;

num = -num;

}

while (num != 0)

{

reverse = (reverse \* 10) + (num % 10);

num /= 10;

}

return sign \* reverse;

}

}

class program

{

static void Main(string[] args)

{

Console.WriteLine("Enter your number ?");

int number = Convert.ToInt32(Console.ReadLine());

Reverse obj = new Reverse();

Console.WriteLine(obj.Mymethod(number));

}

}

}

Merge Two array

using System;

namespace Myapplication

{

class MergeArray

{

public void Mymethod(int[] arr1 , int[] arr2)

{

int totallen = arr1.Length + arr2.Length;

int[] myarr = new int[totallen];

for(int i = 0 , j = 0; i < totallen; i++)

{

if(i < arr1.Length)

{

myarr[i] = arr1[i];

}

else

{

myarr[i] = arr2[j];

j++;

}

}

foreach(int z in myarr)

{

Console.WriteLine(z);

}

}

}

class program

{

static void Main(string[] args)

{

Console.WriteLine("Please enter the Length of your First array :");

int lenf = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Please enter the Length of your second array :");

int lens = Convert.ToInt32(Console.ReadLine());

int[] myarrayf = new int[lenf];

int[] myarrays = new int[lens];

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

{

Console.WriteLine("enter Array Item :");

myarrayf[i] = Convert.ToInt32(Console.ReadLine());

}

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

{

Console.WriteLine("enter Array Item :");

myarrays[i] = Convert.ToInt32(Console.ReadLine());

}

MergeArray obj = new MergeArray();

obj.Mymethod(myarrays , myarrayf);

}

}

}

1. **Write a function to return an array containing all elements common to two given arrays containing distinct positive integers. You should not use any inbuilt methods. You are allowed to use any number of arrays.**  
   The signature of the function is:  
   **int[] f(int[] first, int[] second)**

using System;

namespace Myapplication

{

class Array

{

public void Mymethod(int[] arr1 , int[] arr2)

{

int min;

if(arr1.Length >= arr2.Length)

{

min = arr2.Length;

}

else

{

min = arr1.Length;

}

int[] finalarr = new int[min];

for(int i = 0; i < arr1.Length; i++)

for(int j = 0; j < arr2.Length; j++)

{

if(arr1[i] == arr2[j])

{

finalarr[i] = arr1[i];

}

}

foreach(int z in finalarr)

{

Console.WriteLine("The arrays same item [{0}] : {1}" , z , z );

}

}

}

class program

{

static void Main(string[] args)

{

Console.WriteLine("Please Enter the Length of first array :");

int lenf = Convert.ToInt32(Console.ReadLine());

int[] farr = new int[lenf];

Console.WriteLine("Please Enter the Length of second array :");

int lens = Convert.ToInt32(Console.ReadLine());

int[] sarr = new int[lens];

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

{

Console.WriteLine("Please Enter the first array elements :");

farr [i] = Convert.ToInt32(Console.ReadLine());

}

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

{

Console.WriteLine("Please Enter the second array element :");

sarr[i] = Convert.ToInt32(Console.ReadLine());

}

Array obj = new Array();

obj.Mymethod(farr , sarr);

}

}

}