**C# HANDSON 1**

**NAME – ASHI JAIN**

**BATCH – IBM .NET**

**1.calculateSal**

Read the question carefully and follow the input and output format.

Karen got salary for this month and she spends 20% of her salary for food and 30% of her

salary for travel. If she takes care of other shifts she will get 2% of the salary per day. Given her

salary and the number of shifts she handled. Calculate how much she can save in her pocket

after spending all these?

Input and Output Format :

First line of input consists of an integer, salary. Next line correspond to the number of shifts.

Output consist of an integer, which is saving.

1) Print "Salary too large" when salary is greater than 8000.

2) Print "Shifts too small" when the shift is less than 0.

3) Print "Salary too small" when the salary is less than 0.

Include a function named calculateSal(int salary, int shifts) whose return type is an integer,

which is the saving.

Sample Input 1:

7000

5

Sample Output 1:

4200

Sample Input 2:

8001

Sample Output 2:

Salary too large

**CODE ->**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DemoApp2

{

public class Program

{

int calculateSal(int salary, int shifts)

{

int saving = 0;

saving = (int)((salary \* 0.5) + (salary \* 0.02 \* shifts));

return (saving);

}

static void Main(string[] args)

{

int salary = 0, shifts=0, result=0;

Console.WriteLine("Enter the salary: ");

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

if (salary <= 8000)

{

Console.WriteLine("\nEnter the number of shifts: ");

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

if (shifts < 0)

{

Console.WriteLine("Shifts too small");

}

else if (salary < 0)

{

Console.WriteLine("Salary too small");

}

else

{

var obj = new Program();

result = obj.calculateSal(salary, shifts);

Console.WriteLine("Result: " + result);

}

}

else

{

Console.WriteLine("Salary too large");

}

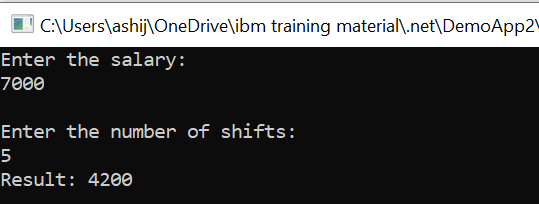
Console.ReadLine();

}

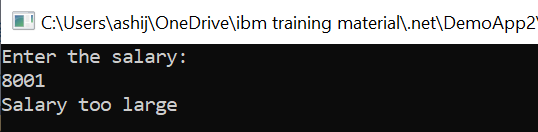
}

}

**OUTPUT 1 ->**



**OUTPUT 2 ->**



**2.Repeated Salary Count**

John is working as a clerk in an organization where N number of people are working. His boss

has asked him to get the count of employees who get same salary. Help him to get the count of

repeated salary.

Include a function named countRepeaters that accepts 2 arguments and returns an int. The

first argument is the input array and the second argument is an int that corresponds to the size

of the array. The function returns an int that corresponds to the number of repeaters.

If the size of the array is negative or if any of the array elements are negative, print “Invalid

Input” and terminate the program.

Input and Output Format:

Input consists of n+1 integers. The first integer corresponds to n, the number of elements in

the array. The next 'n' integers correspond to the elements in the array.

Output consists of an integer that corresponds to the number of repeaters.

Assume that utmost one element in the array would repeat.

Assume that the maximum number of elements in the array is 20.

Sample Input 1:

5

1000

2000

3500

2000

5000

Sample Output 1:

2

Sample Input 2:

-5

Sample Output 2:

Invalid Input

Sample Input 3:

5

1000

-2000

Sample Output 3:

Invalid Input

**CODE ->**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DemoApp4

{

public class Program

{

int CountRepeaters(int[] arr,int n)

{

int i, j, count = 0;

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

if (arr[i] < 0)

{

Console.WriteLine("Invalid Input");

}

}

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

{

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

{

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

{

count+=2;

break;

}

}

}

return count;

}

static void Main(string[] args)

{

int n;

Console.WriteLine("Enter the size of the array: ");

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

if (n > 0)

{

int[] arr1 = new int[n];

Console.WriteLine($"Enter the {n} integers: ");

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

{

arr1[i] = int.Parse(Console.ReadLine());

if(arr1[i]<0)

{

Console.WriteLine("Invalid Input");

}

}

var obj = new Program();

int result = obj.CountRepeaters(arr1, n);

Console.WriteLine("Result: " + result);

}

else

{

Console.WriteLine("Invalid Input");

}

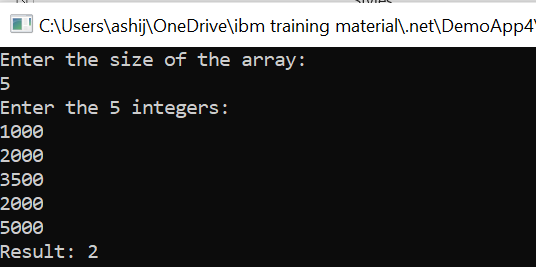
Console.ReadLine();

}

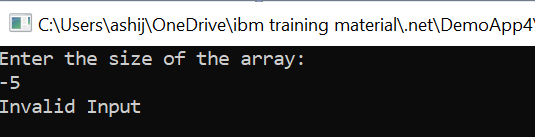
}

}

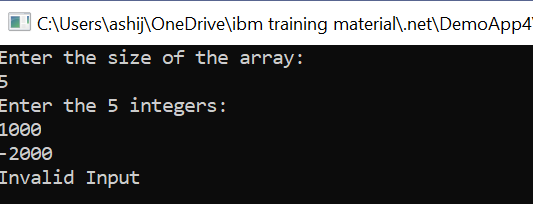
**OUTPUT 1 ->**



**OUTPUT 2 ->**



**OUTPUT 3 ->**



**3.maximumSum**

Read the question carefully and follow the input and output format.

Given an Integer array, find out sum of Even and odd Numbers individually and find the

maximum.

Input and Output Format:

First line of input consists of n, the number of elements. Next n lines correspond to the array

elements. Output consists of maximum of odd and even sum.

1) Print "Invalid array size" when size of the array is a negative number and terminate the

program.

2) Print "Invalid input" when there are any negative numbers available in the input array and

terminate the program.

Include a function named maximumSum(int numbers[], int size) whose return type is an

Integer.

Sample Input 1:

5

12

13

14

15

16

Sample Output 1:

42

Sample Input 2:

-13

Sample Output 2:

Invalid array size

**CODE ->**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DemoApp5

{

public class Program

{

int maximumSum(int[] numbers, int size)

{

int i,sum\_even=0,sum\_odd=0;

if (size < 0)

{

Console.WriteLine("Invalid Input");

}

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

{

if (numbers[i] < 0)

{

Console.WriteLine("Invalid Input");

}

}

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

{

if (i % 2 == 0)

{

sum\_even=sum\_even+numbers[i];

}

else

{

sum\_odd=sum\_odd+numbers[i];

}

}

return Math.Max(sum\_even,sum\_odd);

}

static void Main(string[] args)

{

int size;

Console.WriteLine("Enter the size of the array: ");

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

if (size > 0)

{

int[] number = new int[size];

Console.WriteLine($"Enter the {size} integers: ");

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

{

number[i] = int.Parse(Console.ReadLine());

}

var obj = new Program();

int result = obj.maximumSum(number, size);

Console.WriteLine("Result: " + result);

}

else

{

Console.WriteLine("Invalid array size");

}

Console.ReadLine();

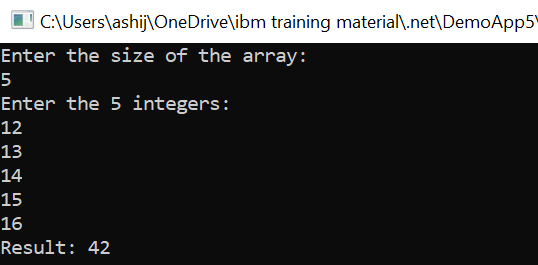
}

}

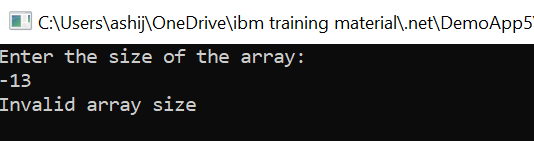
}

}

**OUTPUT 1 ->**



**OUTPUT 2 ->**



**4.Product of Digits**

In a car racing video game, the car is an object. You can drive the car, turn the car, or stop the

car when needed but you need to drive long. You will get money according to the Km you have

travelled. For example if you have travelled 123 km then the product of the km (ie 1\*2\*3 = 6)

would be the amount you win. Write a program to find the product of the digits in the given

input number.

Include a function named productDigits that accepts an integer argument and returns an

integer that corresponds to the product of digits in the integer.

The function returns -1 if the input number is negative or greater than 32767.

If the function returns -1, print Invalid Input.

Input and Output Format:

Input consists of an integer.

Output consists of an integer.

Refer sample output for formatting specifications.

Sample Input 1:

32

Sample Output 1:

6

Sample Input 2:

-67

Sample Output 2:

Invalid Input

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DemoApp6

{

public class Program

{

int productDigits(int num)

{

int product = 1, rev;

while(num != 0)

{

rev = num % 10;

product \*= rev;

num = num / 10;

}

return product;

}

static void Main(string[] args)

{

int num;

Console.WriteLine("Enter the number: ");

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

var obj=new Program();

int result = obj.productDigits(num);

Console.WriteLine("Result: "+result);

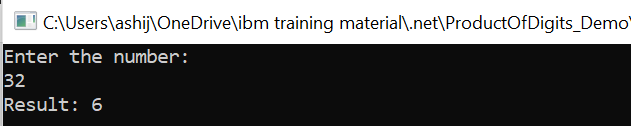
Console.ReadLine();

}

}

}

**OUTPUT 1 ->**



**OUTPUT 2 ->**

