Text

Description automatically generated**Screenshots**

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generated

**Code**

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Homework\_6.\_1

{

internal class Program

{

static int Search(int[] A, int key)

{

bool None = true;

for (int i = 0; A.Length > i; i++)

{

if (A[i] == key)

{

Console.WriteLine("Student #{0} earns 100%\n",i+1);

None = false;

}

}

if (None == true)

{

Console.WriteLine("None of the students earn 100%");

}

return -1;

}

static int integerPowerA(int x, int y)

{

int total = 1;

for (int i = 0; y > i; i++)

{

Console.Write(x);

total \*= x;

if (y != i && y - 1 != i)

{

Console.Write(" \* ");

}

}

Console.Write(" = " + total);

return total;

}

static int integerPowerB(int x, int y)

{

int sum, power;

sum = 1;

power = 1;

for (int i = 0; y >= i; i++)

{

if (y != i)

{

Console.Write(power + " + ");

}

else

{

Console.Write(power);

}

sum += power;

power \*= x;

}

Console.WriteLine(" = " + sum);

return 0;

}

static void Exercise1()

{

int number\_of\_students;

Console.WriteLine("Type the amount of students");

number\_of\_students = Convert.ToInt32(Console.ReadLine());

int[] student\_grades = new int[number\_of\_students];

Console.WriteLine("\nType the student grades");

for (int i = 0; number\_of\_students > i; i++)

{

Console.Write("The grade of the student #{0}",i+1 + " is: ");

student\_grades[i] = Convert.ToInt32(Console.ReadLine());

}

Console.WriteLine();

Search(student\_grades, 100);

Console.ReadLine();

}

static void Exercise2A()

{

int a, b;

Console.WriteLine("The power function works with 2 inputs, a ^ b,\n" +

"A as the base, and B as the exponent. \n\n" +

"-> Assign values for 'a' and 'b' <-\n");

Console.Write("a value: ");

a = Convert.ToInt32(Console.ReadLine());

Console.Write("b value: ");

b = Convert.ToInt32(Console.ReadLine());

Console.WriteLine();

Console.WriteLine(a + "^" + b + " can be broken down in:");

integerPowerA(a, b);

Console.ReadLine();

}

static void Exercise2B()

{

int a, b;

Console.WriteLine("The power function works with 2 inputs, a ^ b,\n" +

"A as the base, and B as the exponent. \n\n" +

"-> Assign values for 'a' and 'b' <-\n");

Console.Write("a value: ");

a = Convert.ToInt32(Console.ReadLine());

Console.Write("b value: ");

b = Convert.ToInt32(Console.ReadLine());

Console.WriteLine();

Console.WriteLine("");

integerPowerB(a, b);

Console.ReadLine();

}

static void Main(string[] args)

{

int option;

Console.WriteLine("Type 1 for Search function \nType 2 for Power function (A) \nType 3 for Power function sum (B)" +

"\n-----------------------------------");

Console.Write("\nOption: ");

option = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("\n");

if (option == 1)

{

Exercise1();

}

else if (option == 2)

{

Exercise2A();

}

else if (option == 3)

{

Exercise2B();

}

else

{

Console.WriteLine("Wrong input");

}

Console.WriteLine();

}

}

}

/\*

static int Search(int[] A, int key)

{

for (int i = 0; A.Length > i; i++)

{

if (A[i] == key)

{

return i;

}

}

return -1;

}

\*/

**Screenshots**Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

**Code**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Reflection.Emit;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Howework\_6.\_2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

private void label5\_Click(object sender, EventArgs e)

{

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

}

private void button1\_Click(object sender, EventArgs e)

{

int User\_Input;

String User\_Input\_Validation = Weekly\_Box.Text;

bool validation = int.TryParse(User\_Input\_Validation, out User\_Input);

double comission, tax, contribution, SSN, total;

if (validation)

{

if (User\_Input > 0)

{

comission = User\_Input \* 0.07;

tax = comission \* 0.18;

contribution = comission \* 0.15;

SSN = comission \* 0.09;

total = comission - tax - contribution - SSN;

Tax\_Box.Text = Convert.ToString(tax);

Retirement\_Box.Text = Convert.ToString(contribution);

SSN\_Box.Text = Convert.ToString(SSN);

Total\_Label.Text = "$" + comission + " - $" + tax + " - $" + contribution + " - $" + SSN + " = $" + total;

}

else

{

MessageBox.Show("Only positive numbers");

}

}

else

{

MessageBox.Show("Numbers only");

}

}

}

}

**Notes**

* Green text / comment is the function requested in 6.1 A
* 6.2 B requests 26 students. I left that variable (26) as Number\_Of\_Students so it can be modified by the user if desired.
* I did a function that shows 2A as option 2.
* 2B function is option 3.
* Both of functions 2A and 2B are done “differently” but “equally”. I changed the logic to display both ways (if statements).
* For 3, final result is given by modifying the values of label 9.
* Message boxes contain concise text.