**05\_Conditional Statements**

**Pr. 01\_Exchange If Greater**

Write an **if**-statement that takes two integer variables a and b and **exchanges** their values if the first one is greater than the second one. As a result print the values a and b, separated by a space. Examples:

|  |  |  |
| --- | --- | --- |
| **a** | **b** | **result** |
| 5 | 2 | 2 5 |
| 3 | 4 | 3 4 |
| 5.5 | 4.5 | 4.5 5.5 |

using System;

class ExchangeIfGreater

{

static void Main()

{

Console.Write("Enter a = ");

double a = double.Parse(Console.ReadLine());

Console.Write("Enter b = ");

double b = double.Parse(Console.ReadLine());

bool grater = (a > b);

bool equal = (a == b);

if (grater)

{

Console.WriteLine(b + " " + a);

}

else if (equal)

{

Console.WriteLine("a = b = " + a);

}

else

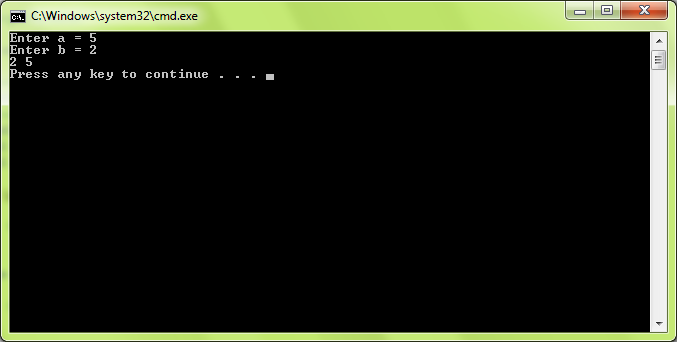
{

Console.WriteLine(a + " " + b);

}

}

}



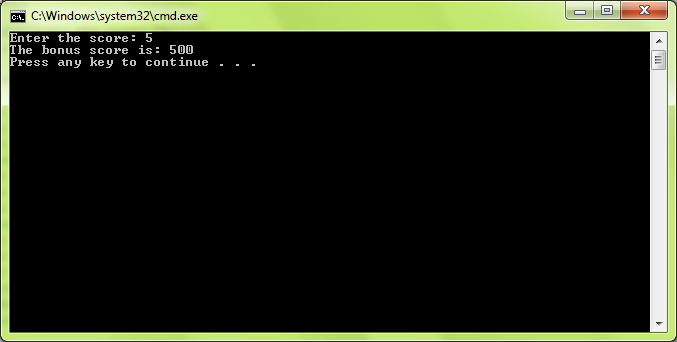
**Pr.02\_Bonus Score**

Write a program that applies bonus score to given score in the range [1…9] by the following rules:

* If the score is between 1 and 3, the program multiplies it by 10.
* If the score is between 4 and 6, the program multiplies it by 100.
* If the score is between 7 and 9, the program multiplies it by 1000.
* If the score is 0 or more than 9, the program prints “invalid score”.

Examples:

|  |  |
| --- | --- |
| **score** | **result** |
| 2 | 20 |
| 4 | 400 |
| 9 | 9000 |
| -1 | invalid score |



## Pr.03.\_Check for a Play Card

Classical play cards use the following signs to designate the card face: 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K and A. Write a program that enters a string and prints “yes” if it is a valid card sign or “no” otherwise. Examples:

|  |  |
| --- | --- |
| **character** | **Valid card sign?** |
| 5 | yes |
| 1 | no |

using System;

class CheckingPlayingCards

{

static void Main()

{

Console.Write("Enter a string: ");

string playingCard = Console.ReadLine();

Console.WriteLine("Is this a valid card sign?");

switch (playingCard)

{

case "2":

case "3":

case "4":

case "5":

case "6":

case "7":

case "8":

case "9":

case "10":

case "J":

case "Q":

case "K":

case "A":

Console.WriteLine("yes"); break;

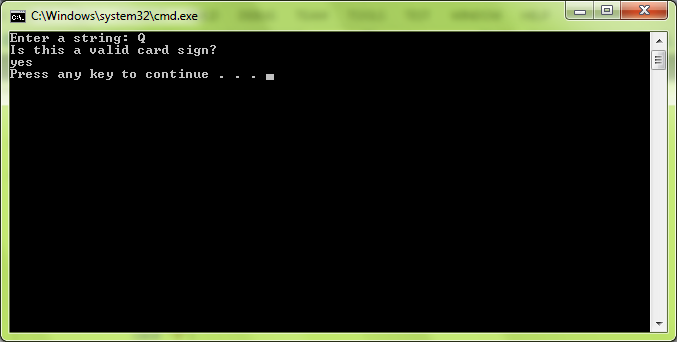
default:

Console.WriteLine("no"); break;

}

}

}



**Pr.04\_Multiplication Sign**

Write a program that shows the sign (+, - or 0) of the product of three real numbers, without calculating it. Use a sequence of **if** operators. Examples:

|  |  |  |  |
| --- | --- | --- | --- |
| **a** | **b** | **c** | **result** |
| 5 | 2 | 2 | + |
| -2 | -2 | 1 | + |
| -2 | 4 | 3 | - |
| 0 | -2.5 | 4 | 0 |

using System;

class MultiplicationSign

{

static void Main()

{

Console.Write("Enter a = ");

double a = double.Parse(Console.ReadLine());

Console.Write("Enter b = ");

double b = double.Parse(Console.ReadLine());

Console.Write("Enter c = ");

double c = double.Parse(Console.ReadLine());

int countMinus = 0;

if (a < 0)

{

countMinus++;

}

if (b < 0)

{

countMinus++;

}

if (c < 0)

{

countMinus++;

}

if (a == 0 | b == 0 | c == 0)

{

Console.WriteLine("The sign of the product of these three numbers is: 0");

}

else if (countMinus % 2 != 0)

{

Console.WriteLine("The sign of the product of these three numbers is: -");

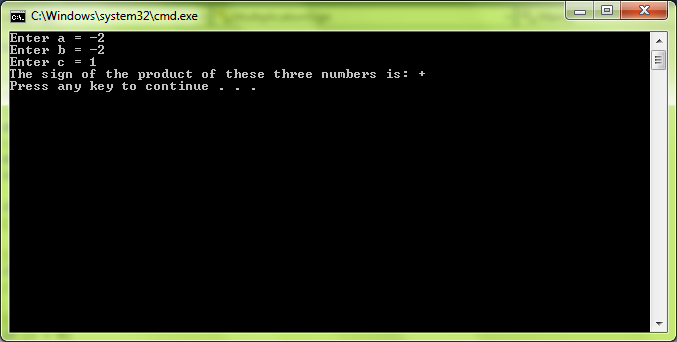
}

else

Console.WriteLine("The sign of the product of these three numbers is: +");

}

}



**Pr.05\_The Biggest Of Three Numbers**

Write a program that finds the **biggest of three numbers**. Examples:

|  |  |  |  |
| --- | --- | --- | --- |
| **a** | **b** | **c** | **biggest** |
| **5** | 2 | 2 | 5 |
| -2 | -2 | **1** | 1 |
| **-0.1** | -0.5 | -1.1 | -0.1 |

using System;

class TheBiggestOf3Numbers

{

static void Main()

{

Console.Write("Enter a = ");

double a = double.Parse(Console.ReadLine());

Console.Write("Enter b = ");

double b = double.Parse(Console.ReadLine());

Console.Write("Enter c = ");

double c = double.Parse(Console.ReadLine());

double biggestNum = int.MinValue;

if (a >= b && a >= c)

{

biggestNum = a;

}

if (b >= a && b >= c)

{

biggestNum = b;

}

if (c >= a && c >= b)

{

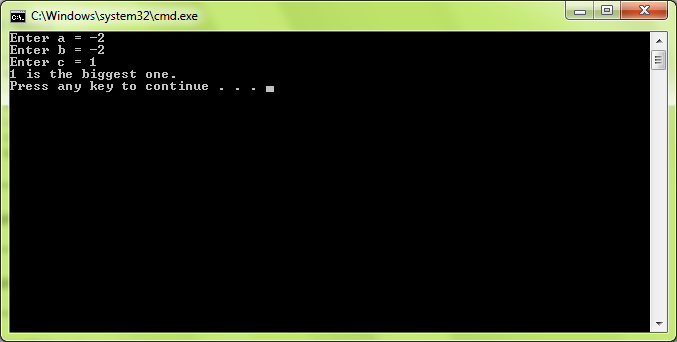
biggestNum = c;

}

Console.WriteLine(biggestNum + " is the biggest one.");

}

}



**Pr.06\_The Biggest Of Five Numbers**

Write a program that finds the **biggest of five numbers** by using only five if statements. Examples:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **a** | **b** | **c** | **d** | **e** | **biggest** |
| **5** | 2 | 2 | 4 | 1 | 5 |
| -2 | -22 | **1** | 0 | 0 | 1 |
| -3 | -0.5 | -1.1 | -2 | **-0.1** | -0.1 |

using System;

class TheBiggestOfFiveNumbers

{

static void Main()

{

Console.Write("Enter a = ");

double a = double.Parse(Console.ReadLine());

Console.Write("Enter b = ");

double b = double.Parse(Console.ReadLine());

Console.Write("Enter c = ");

double c = double.Parse(Console.ReadLine());

Console.Write("Enter d = ");

double d = double.Parse(Console.ReadLine());

Console.Write("Enter e = ");

double e = double.Parse(Console.ReadLine());

double biggestOne = int.MinValue;

if (a >= b && a >= c && a >= d && a >= e)

{

biggestOne = a;

}

if (b >= a && b >= c && b >= d && b >= e)

{

biggestOne = b;

}

if (c >= a && c >= b && c >= d && c >= e)

{

biggestOne = c;

}

if (d >= a && d >= b && d >= c && d >= e)

{

biggestOne = d;

}

if (e >= a && e >= b && e >= c && e >= d)

{

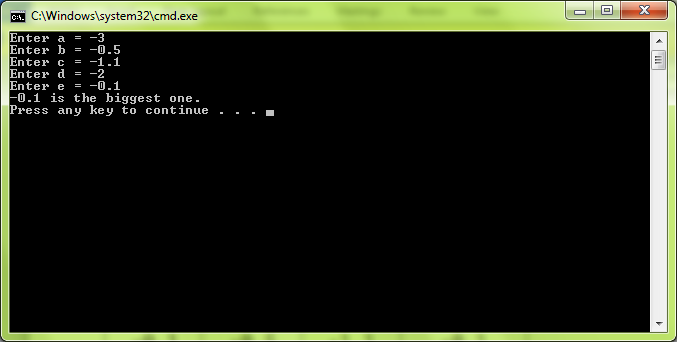
biggestOne = e;

}

Console.WriteLine(biggestOne + " is the biggest one.");

}

}



## Pr.07\_ Sort 3 Numbers with Nested Ifs

Write a program that enters **3 real numbers** and prints them sorted in descending order. Use nested **if** statements. Don’t use arrays and the built-in sorting functionality. Examples:

|  |  |  |  |
| --- | --- | --- | --- |
| **a** | **b** | **c** | **result** |
| 5 | 1 | 2 | 5 2 1 |
| -2 | -2 | 1 | 1 -2 -2 |
| -2 | 4 | 3 | 4 3 -2 |

using System;

class Sort3NumbersWithNestedIfs

{

static void Main()

{

Console.Write("Enter a = ");

double a = double.Parse(Console.ReadLine());

Console.Write("Enter b = ");

double b = double.Parse(Console.ReadLine());

Console.Write("Enter c = ");

double c = double.Parse(Console.ReadLine());

if (a >= b && a >= c)

{

if (b >= c)

{

Console.WriteLine(a + " " + b + " " + c);

}

else if (c >= b)

{

Console.WriteLine(a + " " + c + " " + b);

}

}

else if (b >= a && b >= c)

{

if (a >= c)

{

Console.WriteLine(b + " " + a + " " + c);

}

else if (c >= a)

{

Console.WriteLine(b + " " + c + " " + a);

}

}

else if (c >= a && c >= b)

{

if (a >= b)

{

Console.WriteLine(c + " " + a + " " + b);

}

else if (b >= a)

{

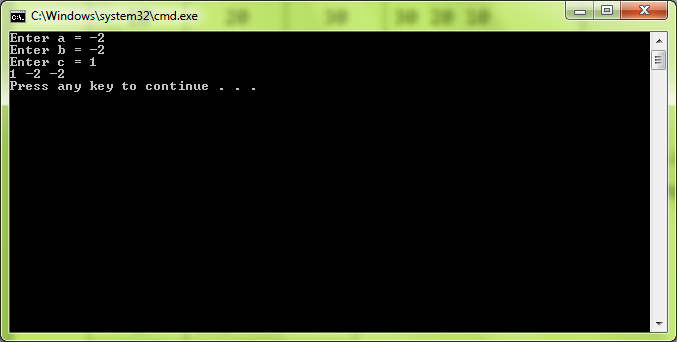
Console.WriteLine(c + " " + b + " " + a);

}

}

}

}



**Pr.08\_Digit As Word**

Write a program that asks for a **digit** (0-9), and depending on the input, **shows the digit as a word** (in English). Print “not a digit” in case of invalid inut. Use a **switch** statement. Examples:

|  |  |
| --- | --- |
| **d** | **result** |
| 2 | two |
| 1 | one |
| 0 | zero |
| 5 | five |
| -0.1 | not a digit |

using System;

class DigitAsWord

{

static void Main()

{

Console.Write("Enter a digit: ");

string digit = Console.ReadLine();

switch (digit)

{

case "1":

Console.WriteLine("one"); break;

case "2":

Console.WriteLine("two"); break;

case "3":

Console.WriteLine("three"); break;

case "4":

Console.WriteLine("four"); break;

case "5":

Console.WriteLine("five"); break;

case "6":

Console.WriteLine("six"); break;

case "7":

Console.WriteLine("seven"); break;

case "8":

Console.WriteLine("eight"); break;

case "9":

Console.WriteLine("nine"); break;

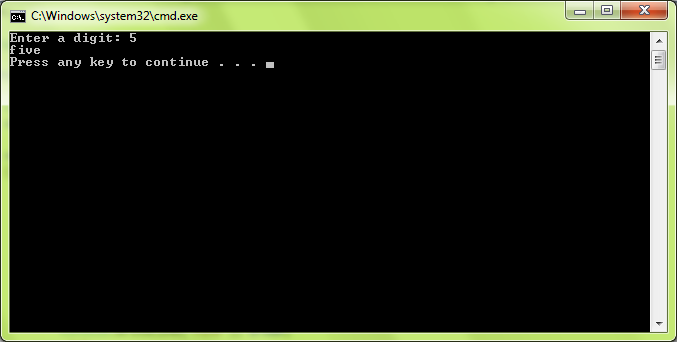
default:

Console.WriteLine("not a digit"); break;

}

}

}



**Pr.09\_Play With Int Double And String**

Write a program that, depending on the user’s choice, inputs an **int**, **double** or **string** variable. If the variable is **int** or **double**, the program increases it by one. If the variable is a **string**, the program appends "**\***" at the end. Print the result at the console. Use **switch** statement. Example:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **program** | **user** |  | **program** | **user** |
| Please choose a type: 1 --> int  2 --> double  3 --> string | 3 | Please choose a type: 1 --> int  2 --> double  3 --> string | 2 |
| Please enter a string: | hello | Please enter a double: | 1.5 |
| hello\* |  | 2.5 |  |

using System;

class PlayWithIntDoubleAndString

{

static void Main()

{

Console.Write(@"Please choose a type:

1 --> int

2 --> double

3 --> string

");

string choice = Console.ReadLine();

switch (choice)

{

case "1":

Console.Write("Please enter an int: ");

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

Console.WriteLine(intNum + 1); break;

case "2":

Console.Write("Please enter a double: ");

double doubleNum = double.Parse(Console.ReadLine());

Console.WriteLine(doubleNum + 1); break;

case "3":

Console.Write("Please enter a string: ");

string notNum = Console.ReadLine();

Console.WriteLine(notNum + "\*"); break;

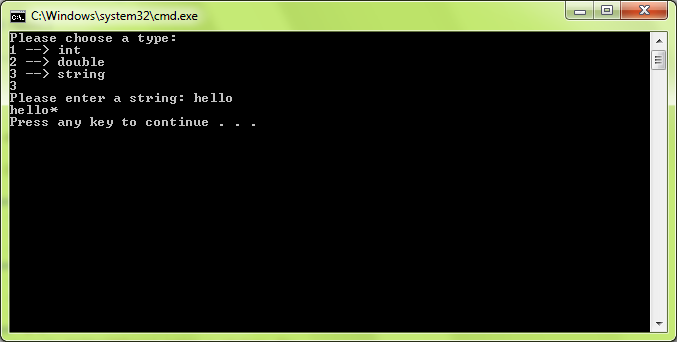
default:

Console.WriteLine("Invalid input."); break;

}

}

}



**Pr.10\_BeerTime**

A beer time is after 1:00 PM and before 3:00 AM. Write a program that **enters a time** in format “hh:mm tt” (an hour in range [01...12], a minute in range [00…59] and AM / PM designator) and prints “**beer time**” or “**non-beer time**” according to the definition above or “**invalid time**” if the time cannot be parsed. Note that you may need to learn how to parse dates and times. Examples:

|  |  |
| --- | --- |
| **time** | **result** |
| 1:00 PM | beer time |
| 4:30 PM | beer time |
| 10:57 PM | beer time |
| 8:30 AM | non-beer time |

using System;

class BeerTime

{

static void Main()

{

string time = Console.ReadLine();

DateTime startBeer = DateTime.Parse("1:00 PM");

DateTime endBeer = DateTime.Parse("3:00 AM");

DateTime beer;

bool result = DateTime.TryParseExact(time, "h:mm tt", new System.Globalization.CultureInfo("en-US"), System.Globalization.DateTimeStyles.None, out beer);

bool beerTime = (startBeer <= beer) || (beer < endBeer);

if (result)

{

if (beerTime)

{

Console.WriteLine("beer time");

}

else

{

Console.WriteLine("non beer time");

}

}

else

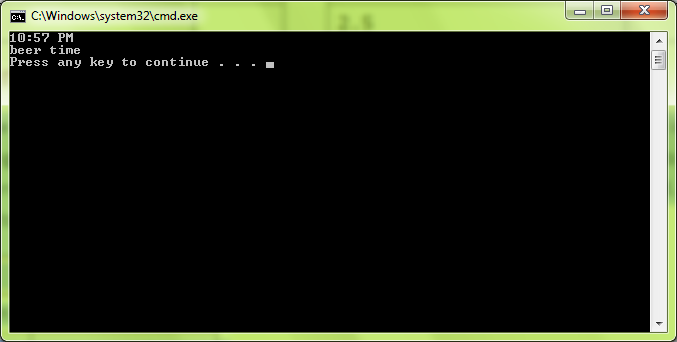
{

Console.WriteLine("Invalid time");

}

}

}



**Pr.11\_Number As Words**

Write a program that **converts a number in the range [0…999] to words**, corresponding to the English pronunciation. Examples:

|  |  |
| --- | --- |
| **numbers** | **number as words** |
| 0 | Zero |
| 9 | Nine |
| 711 | Seven hundred and eleven |

using System;

class NumberAsWords

{

static void Main()

{

Console.Write("Enter a number in the range [0...999]: ");

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

string[] unitsAndHundreds = { "One", "Two", "Three", "Four", "Five", "Six", "Seven", "Eight", "Nine", "Ten", "Eleven", "Twelve", "Thirteen", "Fourteen", "Fifteen", "Sixteen", "Seventeen", "Eighteen", "Nineteen" };

string[] tens = { "Twenty", "Thirty", "Fourty", "Fifty", "Sixty", "Seventy", "Eighty", "Ninety" };

int hundredsDigit = num / 100;

int tensDigit = (num / 10) % 10;

int unitsDigit = num % 10;

if (num < 0 || num > 999)

{

Console.WriteLine("Invalid number");

}

else if (num > 99 && num < 1000)

{

Console.Write("{0} hundred and ", unitsAndHundreds[hundredsDigit - 1]);

if (num % 100 < 19)

{

Console.WriteLine("{0}", unitsAndHundreds[(num % 100) - 1].ToLower());

}

else if (num % 100 > 19)

{

Console.Write("{0} ", tens[tensDigit - 2].ToLower());

if (unitsDigit > 0)

{

Console.WriteLine("{0}", unitsAndHundreds[unitsDigit - 1].ToLower());

}

}

}

else if (19 < num && num < 99)

{

Console.Write("{0} ", tens[tensDigit - 2]);

if (unitsDigit > 0)

{

Console.WriteLine("{0}", unitsAndHundreds[unitsDigit - 1].ToLower());

}

}

else if (num % 100 < 19)

{

Console.WriteLine("{0}", unitsAndHundreds[(num % 100) - 1]);

}

}

}



**Pr.12\_Zero Subset**

We are given 5 integer numbers. Write a program that finds all **subsets of these numbers whose sum is 0**. Assume that repeating the same subset several times is not a problem. Examples:

|  |  |
| --- | --- |
| **numbers** | **result** |
| 3 -2 1 1 8 | -2 + 1 + 1 = 0 |
| 3 1 -7 35 22 | no zero subset |
| 1 3 -4 -2 -1 | 1 + -1 = 0  1 + 3 + -4 = 0  3 + -2 + -1 = 0 |

using System;

class ZeroSubset

{

static void Main()

{

int a, b, c, d, e;

Console.Write("Enter a: ");

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

Console.Write("Enter b: ");

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

Console.Write("Enter c: ");

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

Console.Write("Enter d: ");

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

Console.Write("Enter e: ");

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

int i = 0;

if ((a + b) == 0)

{

Console.WriteLine("{0} + {1} = 0", a, b);

i++;

}

if ((a + c) == 0)

{

Console.WriteLine("{0} + {1} = 0", a, c);

i++;

}

if ((a + d) == 0)

{

Console.WriteLine("{0} + {1} = 0", a, d);

i++;

}

if ((a + e) == 0)

{

Console.WriteLine("{0} + {1} = 0", a, e);

i++;

}

if ((b + c) == 0)

{

Console.WriteLine("{0} + {1} = 0", b, c);

i++;

}

if ((b + d) == 0)

{

Console.WriteLine("{0} + {1} = 0", b, d);

i++;

}

if ((b + e) == 0)

{

Console.WriteLine("{0} + {1} = 0", b, e);

i++;

}

if ((c + d) == 0)

{

Console.WriteLine("{0} + {1} = 0", c, d);

i++;

}

if ((c + e) == 0)

{

Console.WriteLine("{0} + {1} = 0", c, e);

i++;

}

if ((d + e) == 0)

{

Console.WriteLine("{0} + {1} = 0", d, e);

i++;

}

if ((a + b + c) == 0)

{

Console.WriteLine("{0} + {1} + {2} = 0", a, b, c);

i++;

}

if ((a + b + d) == 0)

{

Console.WriteLine("{0} + {1} + {2} = 0", a, b, d);

i++;

}

if ((a + b + e) == 0)

{

Console.WriteLine("{0} + {1} + {2} = 0", a, b, e);

i++;

}

if ((a + c + d) == 0)

{

Console.WriteLine("{0} + {1} + {2} = 0", a, c, d);

i++;

}

if ((a + c + e) == 0)

{

Console.WriteLine("{0} + {1} + {2} = 0", a, c, e);

i++;

}

if ((a + d + e) == 0)

{

Console.WriteLine("{0} + {1} + {2} = 0", a, d, e);

i++;

}

if ((b + c + d) == 0)

{

Console.WriteLine("{0} + {1} + {2} = 0", b, c, d);

i++;

}

if ((b + c + e) == 0)

{

Console.WriteLine("{0} + {1} + {2} = 0", b, c, e);

i++;

}

if ((b + d + e) == 0)

{

Console.WriteLine("{0} + {1} + {2} = 0", b, d, e);

i++;

}

if ((c + d + e) == 0)

{

Console.WriteLine("{0} + {1} + {2} = 0", c, d, e);

i++;

}

if ((a + b + c + d) == 0)

{

Console.WriteLine("{0} + {1} + {2} + {3} = 0", a, b, c, d);

i++;

}

if ((a + b + c + e) == 0)

{

Console.WriteLine("{0} + {1} + {2} + {3} = 0", a, b, c, e);

i++;

}

if ((a + c + d + e) == 0)

{

Console.WriteLine("{0} + {1} + {2} + {3} = 0", a, c, d, e);

i++;

}

if ((a + b + d + e) == 0)

{

Console.WriteLine("{0} + {1} + {2} + {3} = 0", a, b, d, e);

i++;

}

if ((a + b + c + d + e) == 0)

{

Console.WriteLine("{0} + {1} + {2} + {3} + {4} = 0", a, b, c, d, e);

i++;

}

if (i == 0)

{

Console.WriteLine("no zero subset");

}

}

}

