|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **윈도우프로그래밍\_과제\_2주** | **이름 :** | **이갑성** | **학번 :** | **20135151** |

* **실행 결과를 제시하세요**

|  |
| --- |
| 1) 자료형 최대값과 최소값 출력   * double, int, uint 자료형에 대한 최대값과 최소값도 함께 출력될 수 있도록 프로그램을 수정한 후 결과를 제시하시오. |
| class Program{  static void Main(string[] args){  Console.WriteLine(" short : {0} ~ {1} ", short.MinValue, short.MaxValue);  Console.WriteLine(" ushort : {0} ~ {1} ", ushort.MinValue, ushort. MaxValue);  char data = 'a';  Console.WriteLine(" data : {0}, data+1 : {1} ", data, (char)(data + 1));  }  } |
| **[실행결과]** |

|  |
| --- |
| 2) 열거형 |
| class Program  {  enum Suit { Clubs, Hearts, Diamonds, Spades };  enum Color { Red, Green,Blue};  //열거형, 각 멤버의 순서 값은 0부터 시작, 1씩 증가  //특정 순서 값을 원하면 열거형 정의할 때 지정  // enum Suit {Clubs=3, Hearts, Diamonds, Spades}  static void Main(string[] args)  {    Color c = Color.Red;  int j = (int)++c;  Console.WriteLine("Color = " + (Color)j);    Suit trumps = Suit.Hearts;  int i = (int)trumps; //순서값으로 변환, cast 연산  Console.WriteLine("Cardinality of " + trumps + "=" + i);  Console.WriteLine("Cardinality of " + (++trumps) + "=" + (i + 1));  } |
| **[실행결과]** |

|  |
| --- |
| 3) 암시적 자료형 var   * data1 변수의 값을 다음과 같이 변경하는 문장을 추가한 후 실행해 보시오. 만약 오류가 발생하면 오류 메시지를 제시하시오.   data1 = 1161; |
| //GetType() 메소드를 사용하여 자료형 검사  class Program{  static void Main(string[] args){  var data1=34.5f;  var data2 = 34.56;  var data3 = 300;  var data4 = 'a';  //추가하기 data1 = 1161;  Console.WriteLine(" data1 : {0} type : {1} ", data1, data1.GetType());  Console.WriteLine(" data2 : {0} type : {1} ", data2, data2.GetType());  Console.WriteLine(" data3 : {0} type : {1} ", data3, data3.GetType());  Console.WriteLine(" data4 : {0} type : {1} ", data4, data4.GetType());  }  } |
| **[실행결과]** |

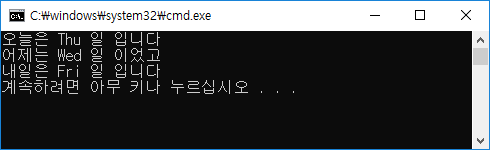
|  |
| --- |
| 4) nullable 형   * ?를 삭제하면 어떤 오류가 발생하는지 확인하세요 * Value 속성을 제거한 후 결과를 확인하세요 |
| class Program{  static void Main(string[] args){  double? num1 = null;  double? num2 = 10.5;  if (num1.HasValue)  Console.WriteLine("num1= {0}", num1.Value);  else  Console.WriteLine("num1 does not have value");  Console.WriteLine("num2= {0}", num2.**Value**); //여기에 있는 Value를 삭제후 출력해보기  }  } |
| **[실행결과]**  **<?>를 삭제했을 때>**    **<Value 속성 삭제했을 때>** |

|  |
| --- |
| 5) 문자열 처리 |
| class Program {  static void Main(string[] args) {  string s1 = "하늘";  string s2 = String.Copy(s1);  string s3 = String.Concat(s1, s2);  Console.WriteLine("문자열 복사 : {0} \n문자열 연결 : {1} ", s2, s3);  string s4 = " 물: 불: 흙: 나무";  string[] spstr = s4.Split(':');  Console.WriteLine("문자열 분리 전 : " + s4);  Console.WriteLine("문자열 분리 후 : ");  for(int i=0; i < spstr.**Length**; i++){  Console.WriteLine(spstr[i]);  }  String str = " Blank String Csharp ";  Console.WriteLine("문자열(길이) = " + str + "(" + str.Length + ")");  Console.WriteLine("공백 제거(Trim)=" + str.Trim());  Console.WriteLine("공백 제거후 문자열 길이(Trime)=" + str.Trim().Length);  Console.WriteLine("왼쪽 공백 제거(TrimStart)= " + str.TrimStart());  Console.WriteLine("왼쪽 공백 제거 후 문자열 길이(TrimStart) = " +  str.TrimStart().Length);  Console.WriteLine("오른쪽 공백 제거(TrimEnd)=" + str.TrimEnd());  Console.WriteLine("오른쪽 공 백 제거 후 문자열 길이(TrimEnd) = " +  str.TrimEnd().Length);  Console.WriteLine("대문자 변환(ToUpper) = " + str.ToUpper());  Console.WriteLine("문자열 치환(Replace) = " + str.Replace('s', 't'));  Console.WriteLine("문자열 검색(IndexOf) = " + str.IndexOf("S"));  Console.WriteLine("문자열 추출(Substring)=" + str.Substring(2, 3));  Console.WriteLine("문자열 삽입(Insert)=" + str.Insert(2, "fun"));  Console.WriteLine("문자열 지우기(Remove)=" + str.Remove(2, 3));  }  } |
| **[실행결과]** |

|  |
| --- |
| 6) escape문자 사용 예 |
| class Program  {  static void Main(string[] args)  {  Console.WriteLine("\t is tab");  Console.WriteLine(" '\t' is tab");  Console.WriteLine(" \'\t\' is tab");  Console.WriteLine(" '\\t\' is tab");  }  } |
| **[실행결과]** |

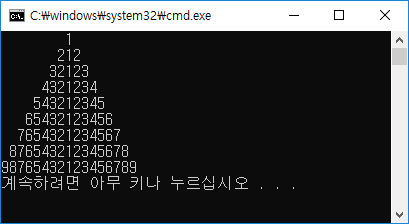
* 프로그램을 작성하세요

1. 일주일의 각 요일을 열거형으로 정의한 후 다음과 같이 출력되는 프로그램을 작성하시오.



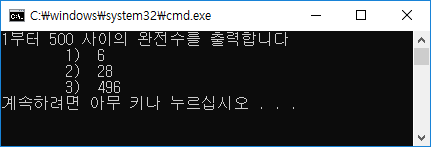
|  |
| --- |
| **프로그램 소스** |
| class Week  {  enum Weekend { Mon, Tue, Wed, Thu, Fri, Sat, Sun };  public static void Main(string[] args)  {  Weekend today = Weekend.Thu;  Console.WriteLine("오늘은 {0} 일 입니다. \n", today);  Console.WriteLine("어제는 {0} 일 입니다. \n", today - 1);  Console.WriteLine("내일은 {0} 일 입니다. \n", today + 1);  }  } |
| **[실행결과]** |

1. 다음과 같은 형태를 출력하는 프로그램을 작성하시오



|  |
| --- |
| **프로그램 소스** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Project01  {  class Pyramid  {  public static void Main(string[] args)  {  for (int i = 1; i < 10; i++)  {  for (int j = 9; j > i; j--)  {  Console.Write(" ");  }  for (int k = i; k >= 1; k--)  {  Console.Write("{0}", k);    }  for (int m = 2; m <= i; m++)  {  Console.Write("{0}", m);  }  Console.Write("\n");  }  }  }  } |
| **[실행결과]** |

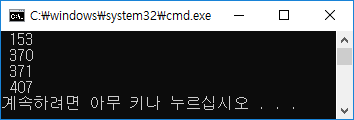
1. 1부터 500사이의 완전수를 출력하는 프로그램을 작성하시오. 완전수 란 자기자신을 제외한 약수의 합이 자기자신과 같은 수이다(예:6=1+2+3)



|  |
| --- |
| **프로그램 소스** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Project01  {  class PerfectNum  {  public static void Main(string[] args)  {  int[] result = new int[100]; //약수를 저장할 배열 생성  int sum; //약수의 합  int index = 0; //배열의 인덱스를 나타내는 변수  Console.WriteLine("1부터 500사이의 완전수를 출력합니다. \n");  for (int i = 1; i <= 500; i++)  {  sum = 0;  for (int j = 1; j < i; j++)  {  if (i % j == 0) //약수 구하는 조건문  {  sum += j;  }  }  if (i == sum)  {  result[index] = sum; //완전수를 저장  index++;  }  }  for (int k = 0; k < index; k++)  {  Console.WriteLine("완전수 {0}) {1} 입니다. \n", k + 1, result[k]);  }  }  }  } |
| **[실행결과]** |

1. 암스트롱수란 3개의 숫자로 구성되며 각 자릿수에 있는 숫자의 세제곱의 합이 자신과 같은 수이다. 100부터 500사이의 암스트롱수를 구하는 프로그램을 작성하시오.

XYZ = X3 + Y3 + Z3



|  |
| --- |
| **프로그램 소스** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace Project01  {  class Armstrong {  public static void Main(string[] args)  {  int sum;  int x, y, z;  int rest;  int index = 1;  for (int i = 100; i <= 500; i++)  {  sum = i;  //백의자리 십의자리 일의자리를 쪼개주는 작업  x = i / 100;  rest = i % 100;  y = rest / 10;  rest = rest % 10;  z = rest;  //암스트롱수인지 확인하는 조건문  if (sum == ((x \* x \* x) + (y \* y \* y) + (z \* z \* z)))  {  Console.WriteLine("암스트롱수 {0}) {1}", index++, sum);  }  }  }  }  } |
| **[실행결과]** |