Fatlum Maliqi / Kampusi Ferizaj

Chapter 3. Operators and Expressions

1. Write an expression that checks whether an integer is odd or even.

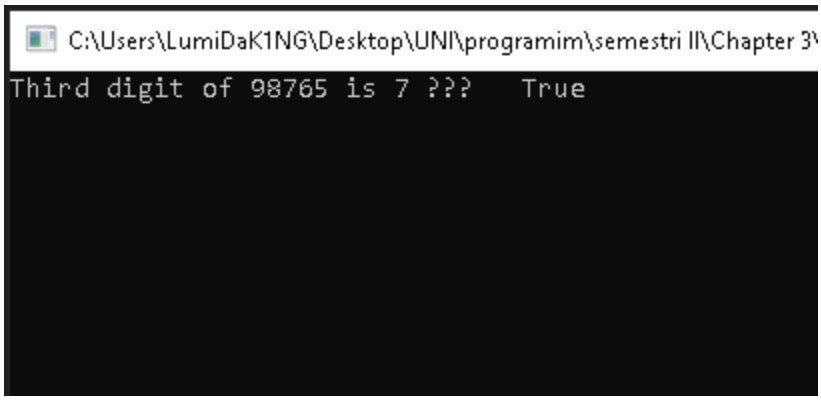
```
using System;

namespace ex1
{
    class Program
    {
        static void Main(string[] args)
        {
            int num = 16;
            bool evenOrodd = num % 2 == 0;
            Console.WriteLine("{0} is even ? {1}", num, evenOrodd);
            Console.ReadKey();
        }
    }
}
```

1. Write a Boolean expression that checks whether a given integer is divisible by both 5 and 7, without a remainder.

```
using System;

namespace ex2
{
   class Program
   {
      static void Main(string[] args)
      {
         int num = 35;
         bool divisible = 35 % 5 == 0 && 35 % 7 == 0;
         Console.WriteLine("Is {0} divisible by 5 and 7 ?   {1}", num, divisible);
         Console.ReadKey();
      }
   }
}
```

2. Write an expression that checks for a given integer if its **third digit** (right to left) is 7.

```
using System;

namespace ex3
{
   class Program
   {
      static void Main(string[] args)
      {
         int num = 98765;
         bool isSeven = (num / 100) % 10 == 7;
         Console.WriteLine("Third digit of {0} is 7 ???   {1}", num, isSeven);
         Console.ReadKey();
      }
   }
}
```

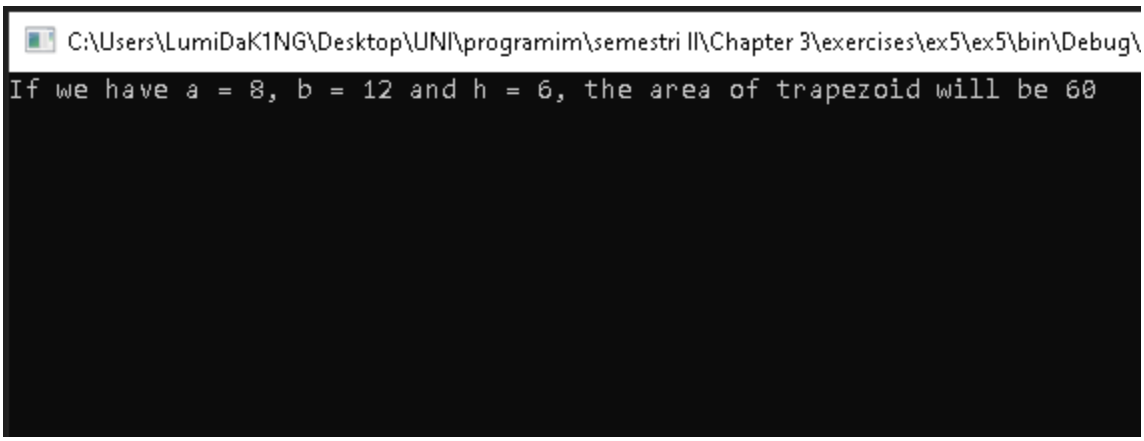


3. Write an expression that checks whether the **third bit** in a given integer is 1 or 0.

```
using System;

namespace ex4
{
   class Program
   {
      static void Main(string[] args)
      {
         int num = 567;
         bool check = ((num >> 3) & 1 )== 1;
         Console.WriteLine("The third bit of {0} is 1 ? {1}", num, check);
         Console.ReadKey();
      }
   }
}
```

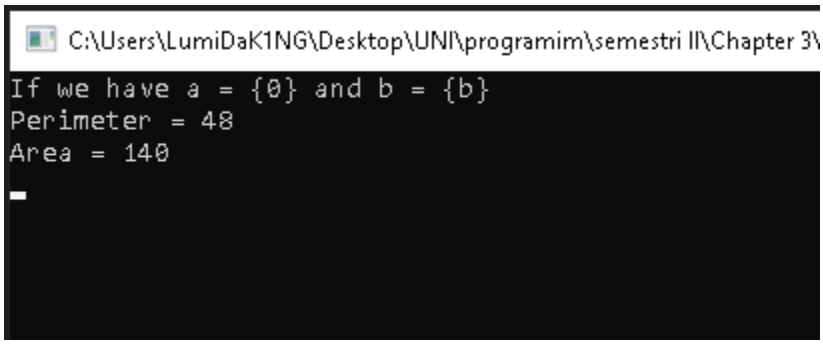
4. Write an expression that calculates the **area of a trapezoid** by given sides **a**, **b** and height **h**.

```
using System;

namespace ex5
{
   class Program
   {
      static void Main(string[] args)
      {
         int a = 8;
         int b = 12;
         int h = 6;
         int area = ((a + b) / 2) * h;
         Console.WriteLine("If we have a = {0}, b = {1} and h = {2}, " +
            "the area of trapezoid will be {3}", a, b, h, area);
         Console.ReadKey();
      }
   }
}
```



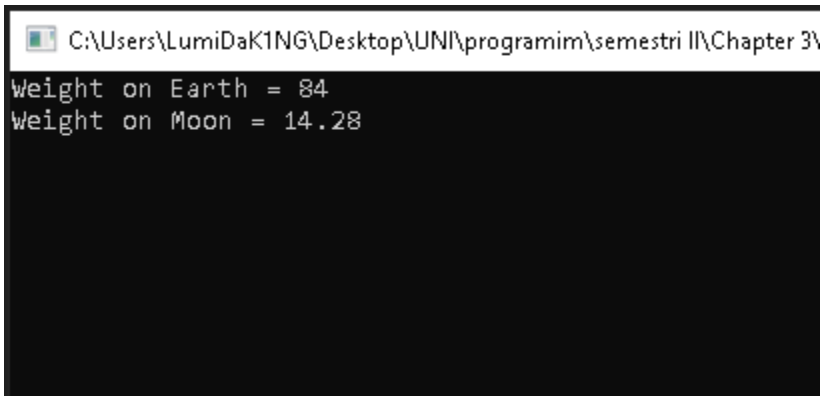
5. Write a program that prints on the console the **perimeter and the area of a rectangle** by given side and height entered by the user.

```
using System;

namespace ex6
{
   class Program
   {
      static void Main(string[] args)
      {
         int a = 14;
         int b = 10;
         int perimeter = (2 * a) + (2 * b);
         int area = (a * b);
         Console.WriteLine("If we have a = {0} and b = {b}");
         Console.WriteLine("Perimeter = " + perimeter);
         Console.WriteLine("Area = " + area);
         Console.ReadKey();
      }
   }
}
```



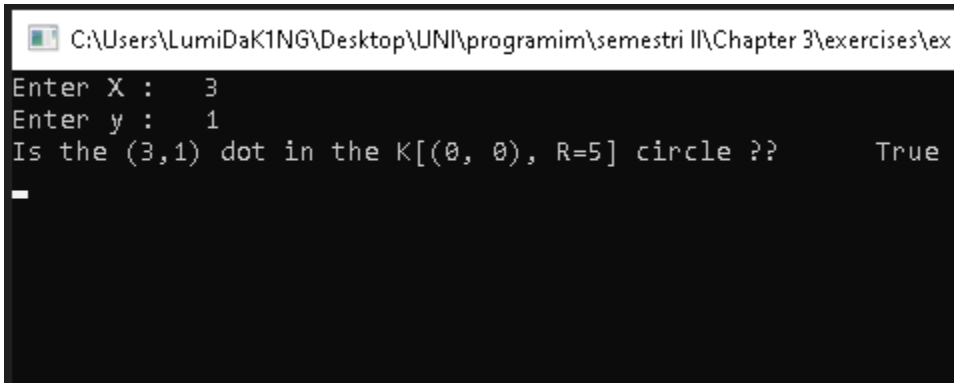
6. The gravitational field of the Moon is approximately 17% of that on the Earth. Write a program that calculates the **weight of a man on the moon** by a given weight on the Earth.

```
using System;

namespace ex7
{
   class Program
   {
      static void Main(string[] args)
      {
         int Eweight = 84;
         double Mweight = (Eweight * 0.17);
         Console.WriteLine("Weight on Earth = " + Eweight);
         Console.WriteLine("Weight on Moon = " + Mweight);
         Console.ReadKey();
      }
   }
}
```



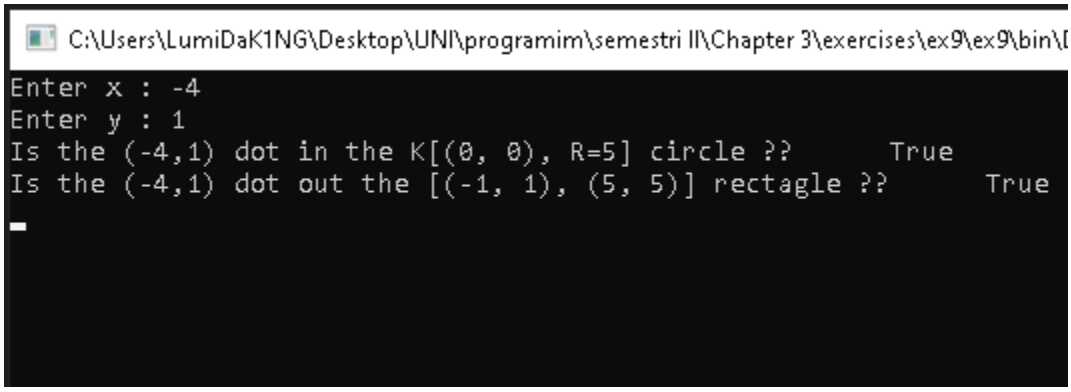
7. Write an expression that checks for a given point {x, y} if it is **within the circle** K[{0, 0}, R=5]. Explanation: the point {0, 0} is the center of the circle and 5 is the radius.

```
using System;

namespace ex8
{
  class Program
  {
    static void Main(string[] args)
    {
      Console.Write("Enter X :   ");
      int x = Convert.ToInt32(Console.ReadLine());
      Console.Write("Enter y :   ");
      int y = Convert.ToInt32(Console.ReadLine());
      bool check = (x < 5) && (y < 5);
      Console.WriteLine("Is the ({0},{1}) dot in the K[(0, 0), R=5] circle ??     {2}", x, y, check);
      Console.ReadKey();
    }
  }
}
```



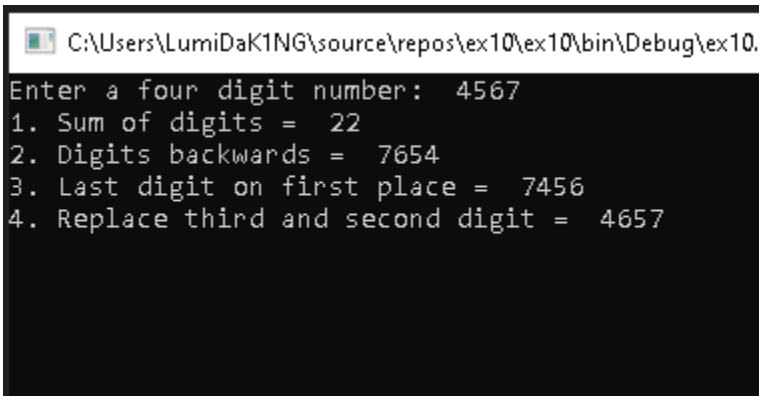
8. Write an expression that checks for given point {x, y} if it is **within the circle** K[{0, 0}, R=5] and **out of the rectangle** [{-1, 1}, {5, 5}]. Clarification: for the rectangle the lower left and the upper right corners are given.

```
using System;

namespace ex9
{
  class Program
  {
    static void Main(string[] args)
    {
      Console.Write("Enter x : ");
      int x = Convert.ToInt32(Console.ReadLine());
      Console.Write("Enter y : ");
      int y = Convert.ToInt32(Console.ReadLine());
      bool CircleCheck = (x < 5) && (y < 5);
      bool RectCheck = (x >= 5 || x <= -1) & (y <= 5 || y >= 1));
      Console.WriteLine("Is the ({0},{1}) dot in the K[(0, 0), R=5]" +
        " circle ??     {2}", x, y, CircleCheck);
      Console.WriteLine("Is the ({0},{1}) dot out the [(-1, 1), (5, 5)]" +
        " rectagle ??     {2}", x, y, RectCheck);
      Console.ReadKey();
    }
  }
}
```



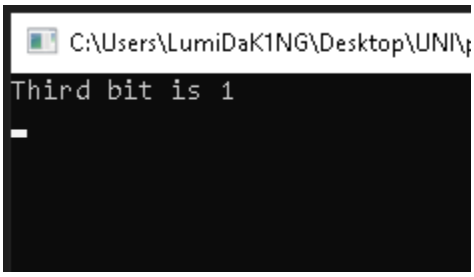
10. Write a program that takes as input a **four-digit number** in format **abcd** (e.g. 2011) and performs the following actions:
    - Calculates the sum of the digits (in our example 2+0+1+1 = 4).
    - Prints on the console the number in reversed order: **dcba** (in our example 1102).
    - Puts the last digit in the first position: **dabc** (in our example 1201).
    - Exchanges the second and the third digits: **acbd** (in our example 2101).

```
using System;

namespace ex10
{
   class Program
   {
      static void Main(string[] args)
      {
         Console.Write("Enter a four digit number:  ");
         int number = Convert.ToInt32(Console.ReadLine());
         int a = number / 1000;
         int b = (number / 100) % 10;
         int c = (number / 10) % 10;
         int d = number % 10;
         Console.WriteLine("1. Sum of digits =  {0}", a + b + c + d);
         Console.WriteLine("2. Digits backwards =  {3}{2}{1}{0}", a, b, c, d);
         Console.WriteLine("3. Last digit on first place =  {3}{0}{1}{2}", a, b, c, d);
         Console.WriteLine("4. Replace third and second digit =  {0}{2}{1}{3}", a, b, c, d);
         Console.ReadKey();
      }
   }
}
```



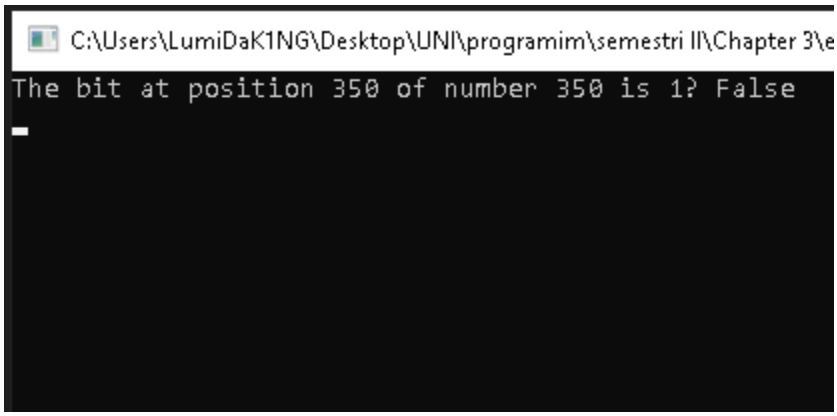
11. We are given number **n** and position **p**. Write a sequence of operations that prints the value of **the bit on the position p** in the number (0 or 1). Example: **n**=35, **p**=5 -> 1. Another example: n=35, **p**=6 -> 0.

```
using System;

namespace ex11
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = 69;
         int p = 6;
         int i = 1;
         int check = i << p;
         Console.WriteLine((n & check) != 0 ? "Third bit is 1" : "Third bit is 0");
         Console.ReadKey();
      }
   }
}
```



12 Write a Boolean expression that checks if the bit on position **p** in the integer **v** has the value 1. Example v=5, **p**=1 -> **false**.

```
using System;

namespace ex12
{
   class Program
   {
      static void Main(string[] args)
      {
         int v = 350;
         int p = 350;
         int check = 1 << p;
         Console.WriteLine("The bit at position {0} of number {1} " +
            "is 1? {2}", p, v, ((v & check) != 0 ? "True":"False"));
         Console.ReadKey();
      }
   }
}
```



12. We are given the number **n**, the value **v** (**v** = 0 or 1) and the position **p**. write a sequence of operations that changes the value of **n**, so the bit on the position **p** has the value of **v**. Example: n=35, p=5, v=0 -> n=3. Another example: n=35, p=2, v=1 -> n=39.

```
using System;

namespace ex13
{
   class Program
   {
      static void Main(string[] args)
      {
         int n = 35;
         int v = 0;
         int p = 5;
         n = (v == 0) ? n = n & (~(1 << p)) : n = n | (1 << p);
         Console.WriteLine(n);
         Console.ReadKey();
      }
   }
}
```

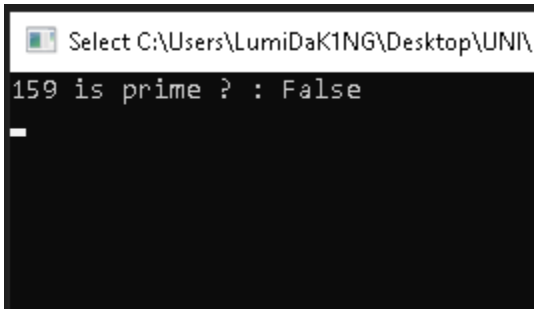
13. Write a program that checks if a given number **n** (1 < **n** < 100) is a **prime number** (i.e. it is divisible without remainder only to itself and 1).

```
using System;

namespace ex14
{
   class Program
   {
      static void Main(string[] args)
      {
         int number = 159;
         bool isPrime = true;
         if (number > 2)
            for (int i = 2; i <= Math.Ceiling(Math.Sqrt(number)); ++i)
            {
               if (number % i == 0) isPrime = false;
            }
         Console.WriteLine("{0} is prime ? : {1}", number, isPrime);
         Console.ReadKey();
      }
   }
}
```

c

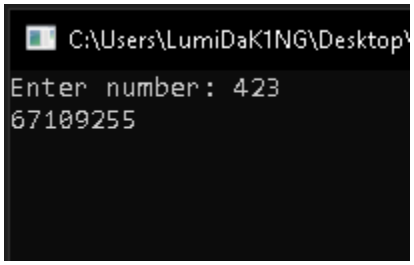
14. * Write a program that **exchanges the values of the bits** on positions 3, 4 and 5 with bits on positions 24, 25 and 26 of a given 32-bit unsigned integer.

```
using System;

namespace ex14
{
   class Program
   {
      static void Main(string[] args)
      {
         int number = 159;
         bool isPrime = true;
         if (number > 2)
            for (int i = 2; i <= Math.Ceiling(Math.Sqrt(number)); ++i)
            {
               if (number % i == 0) isPrime = false;
            }
         Console.WriteLine("{0} is prime ? : {1}", number, isPrime);
         Console.ReadKey();
      }
   }
}
```



15. * Write a program that exchanges the values of the bits on positions 3, 4 and 5 with bits on positions 24, 25 and 26 of a given 32-bit unsigned integer.

```
using System;

namespace ex15
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter number: ");
            int v = Convert.ToInt32(Console.ReadLine());
            int mask = 1 << 3;
            int bitAt3 = (v & mask) != 0 ? 1 : 0;
            mask = 1 << 4;
            int bitAt4 = (v & mask) != 0 ? 1 : 0;
            mask = 1 << 5;
            int bitAt5 = (v & mask) != 0 ? 1 : 0;
            mask = 1 << 24;
            int bitAt24 = (v & mask) != 0 ? 1 : 0;
            mask = 1 << 25;
            int bitAt25 = (v & mask) != 0 ? 1 : 0;
            mask = 1 << 26;
            int bitAt26 = (v & mask) != 0 ? 1 : 0;

            v = (bitAt3 == 0) ? v = v & (~(1 << 24)) : v = v | (1 << 24);
            v = (bitAt4 == 0) ? v = v & (~(1 << 25)) : v = v | (1 << 25);
            v = (bitAt5 == 0) ? v = v & (~(1 << 26)) : v = v | (1 << 26);
            v = (bitAt24 == 0) ? v = v & (~(1 << 3)) : v = v | (1 << 3);
            v = (bitAt25 == 0) ? v = v & (~(1 << 4)) : v = v | (1 << 4);
            v = (bitAt26 == 0) ? v = v & (~(1 << 5)) : v = v | (1 << 5);

            Console.WriteLine(v);
            Console.ReadKey();
        }
    }
}
```



16. * Write a program that **exchanges bits** {p, p+1, ..., p+k-1} with bits {q, q+1, ..., q+k-1} of a given 32-bit unsigned integer.

```
using System;

namespace ex16
{
   class Program
   {
      private static uint ModifyNumber(uint number, int p, int q, int k)
      {
         int[] pBits = new int[k];
         int[] qBits = new int[k];

         for (int position = p, i = 0; i < pBits.Length; position++, i++)
         {
            pBits[i] = PthBit(number, position);
         }

         for (int position = q, i = 0; i < qBits.Length; position++, i++)
         {
            qBits[i] = PthBit(number, position);
         }

         for (int position = p, i = 0; i < qBits.Length; position++, i++)
         {
            number = ModifiedNumber(number, position, qBits[i]);
         }

         for (int position = q, i = 0; i < pBits.Length; position++, i++)
         {
            number = ModifiedNumber(number, position, pBits[i]);
         }
         return number;
      }

      private static int PthBit(uint number, int position)
      {
         uint pthBit = (number >> position) & 1;
         return (int)pthBit;
      }

      private static uint ModifiedNumber(uint number, int position, int bitValue)
      {
         uint actualP = (uint)bitValue << position;
         number = number & (~((uint)1 << position));
         uint result = number | actualP;
         return result;
```

```
    }

    static void Main(string[] args)
    {
      Console.Write("Enter number: ");
      uint number = uint.Parse(Console.ReadLine());
      Console.Write("Enter p: ");
      int p = int.Parse(Console.ReadLine());
      Console.Write("Enter q: ");
      int q = int.Parse(Console.ReadLine());
      Console.Write("Enter k: ");
      int k = int.Parse(Console.ReadLine());


      if (p > q)
      {
        int oldValue = p;
        p = q;
        q = oldValue;
      }

      if (p + k >= q)
      {
        k += p - q - 1;
        q += p + k + 1;
      }

      number = ModifyNumber(number, p, q, k);

      Console.WriteLine(number);
      Console.ReadKey();
    }
  }
}
```