|  |  |
| --- | --- |
| Задача 11. Стойност на бит на дадена позиция | |
| **Условие**  Дадено е число n и позиция p. Напишете поредица от операции, които да отпечатат стойността на бита на позиция p от числото n (0 или 1). Пример: n=35, p=5 -> 1. Още един пример: n=35, p=6 -> 0. | |
| **Описание на входа**  Входните данни се въвеждат на два отделни реда, съответно стойностите на n и p. | |
| **Описание на изхода**  Изходните данни се извеждат като единствено една цифра, стойност на бит (1 или 0). | |
| **Анализ на задачата**  Използваме оператора >>, с който преместваме бита на позиция p на нулева позиция. След това използваме операцията & с 1 за зануляване на останалите битове, освен вече преместения бит. Получената стойност запазваме в променливата bitValue, която извеждаме на екрана. | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace \_11.BitInPositionValue  {  class BitInPositionValue  {  static void Main(string[] args)  {  int n = int.Parse(Console.ReadLine());  int p = int.Parse(Console.ReadLine());  byte bitValue = (byte)(n >> p & 1);  Console.WriteLine(bitValue);  }  }  } | |
| **Тестове**  Интересните случаи за тестване са следните:   * Числото нула * Положително цяло число * Отрицателно цяло число * Наличие на бит 1 на дадена позиция от числото * Наличие на бит 0 на дадена позиция от числото | |
| **Вход** | **Изход** |
| 0  0 | 0 |
| **Вход** | **Изход** |
| 1  0 | 1 |
| **Вход** | **Изход** |
| 4  1 | 0 |
| **Вход** | **Изход** |
| 19  1 | 1 |
| **Вход** | **Изход** |
| -19  4 | 0 |
| **Вход** | **Изход** |
| 255  3 | 1 |
| **Вход** | **Изход** |
| -255  5 | 0 |
| **Вход** | **Изход** |
| 356  3 | 0 |
| **Вход** | **Изход** |
| -356  4 | 1 |
| **Вход** | **Изход** |
| 1000  8 | 1 |