Задание 1

|  |
| --- |
| #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  unsigned int value; int i;  const unsigned int mask = 1 << 31; //делается маска на 32 бита, самый старший бит = 1  cout << "Введите целое число ";  cin >> value;  cout << "Двоичный вид: ";  for (i = 1; i <= 32; i++)  {  putchar(mask & value ? '1' : '0');  value <<= 1;  if (i % 8 == 0) putchar(' '); //пробел после 8 битов  }  }  Вывод программы: |

Задание 2

|  |
| --- |
| #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  int A = 150; char tmp[33];  \_itoa\_s(A, tmp, 2); //конвертируем в двоичное представление  cout << " Число А: " << tmp << endl; //выводим  \_itoa\_s(0x24, tmp, 2); //делаем маску из числе 36(д.с) и переводим её в двоичный 100100  cout << " Маска для А: " << tmp << endl; //вывод маски  \_itoa\_s(A | 0x24, tmp, 2); // операция или  cout << " Результат: " << tmp << endl << endl;  } |

Изменённый вариант:

|  |
| --- |
| #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  int A = 150; char tmp[33];  \_itoa\_s(A, tmp, 2); //конвертируем в двоичное представление  cout << " Число А: " << tmp << endl; //выводим  \_itoa\_s(0x88, tmp, 2); //делаем маску из числе 136(д.с) и переводим её в двоичный 10001000  cout << " Маска для А: " << tmp << endl; //вывод маски  \_itoa\_s(A | 0x88, tmp, 2); // операция или  cout << " Результат: " << tmp << endl << endl;  } |

Задание 3

|  |
| --- |
| #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  int A; char tmp[33];  cout << "Введите число ";  cin >> A;  \_itoa\_s(A, tmp, 2);  cout << "Число в двоичном виде = " << tmp << endl;  if ((A & 7) == 0)  cout << "Число кратно 8" << endl;  else  cout << "Число не кратно 8" << endl;  } |

Изменённый вариант:

|  |
| --- |
| #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  int A;  char tmp[33];  cout << "Введите число ";  cin >> A;  \_itoa\_s(A, tmp, 2);  cout << "Число в двоичном виде = " << tmp << endl;  if ((A & 3) == 0) // последние два бита должны быть равны 0  cout << "Число кратно 4" << endl;  else  cout << "Число не кратно 4" << endl;  } |

Задание 4

|  |
| --- |
| #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  char tmp[33];  int A, B;  int maskA = 14; // маска для извлечения 3 битов (01110 в двоичном виде)  int maskB = ~maskA >> 1; // маска для очищения 3 битов в числе B  cout << "Первое число А=";  cin >> A;  cout << "Второе число В=";  cin >> B;  \_itoa\_s(A, tmp, 2);  cout << "A=" << tmp << endl;  \_itoa\_s(B, tmp, 2);  cout << "B=" << tmp << endl;  \_itoa\_s(maskA, tmp, 2);  cout << "Маска для А: " << tmp << endl;  \_itoa\_s((A & maskA) >> 1, tmp, 2); // извлечение 3 битов из A и сдвиг вправо на 1 бит, затем преобразование в строку  cout << "Выделенные биты А: " << tmp << endl; // вывод извлеченных битов из A  \_itoa\_s(maskB, tmp, 2); //  cout << "Маска для В: " << tmp << endl;  \_itoa\_s(B & maskB, tmp, 2); // Очистка 3 битов в B с помощью маски B и преобразование в строку  cout << " Очищены биты в B: " << tmp << endl; // Вывод числа B с очищенными битами  \_itoa\_s(((B & maskB) | ((A & maskA) >> 1)), tmp, 2); // вставка извлеченных битов из A в B и преобразование в строку  cout << " Результат B=" << tmp << endl;  } |

Вариант 12

|  |
| --- |
| 1. Ввести целое число **A**. Инвертировать все биты с 4 по 8, включая эти биты. Вывести полученное число. |

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  setlocale(LC\_CTYPE, "Russian");  int A;  char tmp[33];  cout << "Введите целое число A: ";  cin >> A;  \_itoa\_s(A, tmp, 2);  cout << "Число A в двоичной системе =" << tmp << endl;  int mask = 0b11111000;  A = A ^ mask;  \_itoa\_s(A, tmp, 2);  cout << "Число после инвертирования битов с 4 по 8: " << A << "\nдвоичное представление:\n" << tmp << endl;  return 0;  } |

|  |
| --- |
| 2. Установить в 0 **n** битов в числе **А** вправо от позиции **p**, заменить ими **m** битов числа **В**, начиная с позиции **q**. |

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "Russian");  int A, B, n, p, q,m;  char tmp[33];  cout << "Введите целое число A: ";  cin >> A;  cout << "Введите целое число B: ";  cin >> B;  cout << "Сколько битов числа A нужно установить в 0? ";  cin >> n;  cout << "Введите позицию p в числе A, вправо от которой нужно установить в 0 биты: ";  cin >> p;  cout << "Сколько битов числа B нужно заменить? ";  cin >> m;  cout << "Введите позицию q в числе B, начиная с которой нужно заменить биты: ";  cin >> q;  \_itoa\_s(A, tmp, 2);  cout << "Число A в двоичной системе: " << tmp << endl;  \_itoa\_s(B, tmp, 2);  cout << "Число B в двоичной системе: " << tmp << endl;  // Создание маски для установки n битов в 0 влево от позиции p  int maskA = ((1 << n) - 1) << (p - n);  \_itoa\_s(maskA, tmp, 2);  cout << "Маска A: " << tmp << endl;  // Установка n битов в 0 в числе A  A = A & ~maskA;  \_itoa\_s(A, tmp, 2);  cout << "Число A после установки битов в 0: " << tmp << endl;  // Извлечение n битов из числа A  int extractedBits = (A & maskA) >> p;  \_itoa\_s(extractedBits, tmp, 2);  cout << "Извлеченные биты из числа A: " << tmp << endl;  // Создание маски для замены m битов в числе B начиная с позиции q  int maskB = ((1 << m) - 1) << q;  \_itoa\_s(maskB, tmp, 2);  cout << "Маска B: " << tmp << endl;  // Очистка m битов в числе B и замена их извлеченными битами  B = (B & ~maskB) | (extractedBits << (q - 1));  \_itoa\_s(B, tmp, 2);  cout << "Число B после замены битов: " << tmp << endl;  return 0;  } |

Допы

Вариант 9(12-3)

|  |
| --- |
| 1. Извлечь 5 битов числа **A**, начиная с третьего по счету справа, и вставить их в число **B**, начиная со 2 по счету справа. |

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "Russian");  int A, B;  char tmp[33];  cout << "Введите число A: ";  cin >> A;  cout << "Введите число B: ";  cin >> B;  \_itoa\_s(A, tmp, 2);  cout << "Число A в двоичном виде: " << tmp << endl;  \_itoa\_s(B, tmp, 2);  cout << "Число B в двоичном виде: " << tmp << endl;  // извлечь 5 битов из числа A, начиная с 3-го бита справа  int extractedBits = (A >> 3) & ((1 << 5) - 1); // сдвигаем A вправо на 3 и маскируем 5 бит  \_itoa\_s(extractedBits, tmp, 2);  cout << "Извлечённые 5 битов из числа A: " << tmp << endl;  // 2. Очистить (установить в 0) соответствующие 5 битов в числе B, начиная со 2-го бита справа  int maskB = ~(((1 << 5) - 1) << 2); // Создаём маску для очистки 5 битов в числе B  B = B & maskB; // Очищаем 5 битов в числе B  // 3. Вставить извлечённые биты в число B, начиная со 2-го бита справа  B = B | (extractedBits << 2); // Вставляем извлечённые биты в очищенное место  // Показать итоговое число B в двоичном формате  \_itoa\_s(B, tmp, 2);  cout << "Число B после вставки: " << tmp << endl;  return 0;  } |

|  |
| --- |
| Установить в 0 **n** битов в числе **А** вправо от позиции **p**. |

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "RU");  int A;  int n, p;  char str[33];  cout << "Введите A: ";  cin >> A;  cout << "Введите n(количество битов, котрые нужно установить в 0): ";  cin >> n;  cout << "Введите p(вправо от позиции p): ";  cin >> p;  \_itoa\_s(A, str, 2);  cout << "Число A до операций:" << endl << str << endl;  // маска с n единицами внизу и сдвиг влево на p позиций  int mask = ~((1 << n) - 1) << (p - n); // Маска, которая ставит n единиц начиная с позиции p  // обнуляем эти биты с помощью побитового И с инвертированной маской  A &= mask; // обнуляем биты, которые соответствуют маске  \_itoa\_s(A, str, 2);  cout << "Число A после операций: " << A << endl;  cout << "Число A после операций в двоичном виде:" << endl << str << endl;  return 0;  } |

Вариант 3 (12+7)

|  |
| --- |
| 1. Ввести целое число **A**. Инвертировать все биты с 2 по 14, включая эти биты. Вывести результат. |

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "Russian");  int A;  char tmp[33];  cout << "Введите целое число A: ";  cin >> A;  cout << "Начальное значение A в двоичном виде: ";  \_itoa\_s(A, tmp, 2);  cout << tmp << endl;  // Создание маски для битов с 2 по 14  int mask = ((1 << (14 - 2 + 1)) - 1) << 2;  // Инвертирование битов с 2 по 14  A ^= mask;  cout << "Число A после инвертирования битов с 2 по 14: ";  \_itoa\_s(A, tmp, 2);  cout << tmp << endl;  return 0;  } |

|  |
| --- |
| Инвертировать **n** битов в числе **А** влево от позиции **p**, заменить ими **m** битов числа **В**, начиная с позиции **q**. |

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "RU");  int A, B;  int n, p, m, q;  char buff[33];  cout << "Введите число A: ";  cin >> A;  cout << "Введите число B: ";  cin >> B;  cout << "Введите количество битов для инверсии в A (n): ";  cin >> n;  cout << "Введите позицию (p), начиная с которой нужно инвертировать биты в A: ";  cin >> p;  cout << "Введите количество битов для замены в B (m): ";  cin >> m;  cout << "Введите позицию (q), начиная с которой нужно заменить биты в B: ";  cin >> q;  \_itoa\_s(A, buff, 2);  cout << "Число A в двоичной системе до изменений: " << buff << endl;  \_itoa\_s(B, buff, 2);  cout << "Число B в двоичной системе до изменений: " << buff << endl;  // 1. Инвертируем n битов в числе A вправо от позиции p  int maskA = ((1 << n) - 1); // Маска с n единицами  maskA <<= (p - n); // Сдвигаем маску влево, чтобы она начиналась с позиции p  A ^= maskA; // Инвертируем n битов в A  // 2. Извлекаем n инвертированных битов из числа A  int extractedBitsA = (A >> (p - n)) & ((1 << n) - 1); // Извлекаем n битов из числа A  // 3. Создаем маску для очистки m битов в числе B, начиная с позиции q  int maskB = ((1 << m) - 1) << q; // Маска для m битов в B, начиная с позиции q  B &= ~maskB; // Очищаем m битов в B  // 4. Вставляем извлечённые инвертированные биты из числа A в число B  B |= (extractedBitsA << q); // Вставляем инвертированные биты в число B  \_itoa\_s(A, buff, 2);  cout << "Число A после инверсии: " << buff << endl;  \_itoa\_s(B, buff, 2);  cout << "Число B после замены битов: " << buff << endl;  cout << "Число B после изменений в десятичной системе: " << B << endl;  return 0;  } |

Вариант 7 (12-5)

|  |
| --- |
| 1. Извлечь 4 бита числа **A**, начиная с пятого по счету справа, и добавить их к числу **B** справа. |

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "ru");  int A, B;  char tmp[33];  cout << "Введите целое число A: ";  cin >> A;  cout << "Введите целое число B: ";  cin >> B;  \_itoa\_s(A, tmp, 2);  cout << "Число A в двоичной системе: " << tmp << endl;  \_itoa\_s(B, tmp, 2);  cout << "Число B в двоичной системе: " << tmp << endl;  // Извлечь 4 бита числа A, начиная с пятого справа  int extractedBits = (A >> 4) & 0b1111; // Сдвигаем вправо на 4 и берем 4 младших бита  \_itoa\_s(extractedBits, tmp, 2);  cout << "Извлечённые 4 бита из A: " << tmp << endl;  // Добавить извлечённые биты к числу B справа  B = (B << 4) | extractedBits; // Сдвиг B на 4 бита влево и добавление битов A  \_itoa\_s(B, tmp, 2);  cout << "Число B после добавления битов: " << tmp << endl;  return 0;  } |

|  |
| --- |
| Установить в 1 **n** битов в числе **А** влево от позиции **p**, заменить ими **m** битов числа **В**, начиная с позиции **q**. |

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  setlocale(LC\_CTYPE, "ru");  int A, B, n, p, m, q;  char tmp[33];  cout << "Введите число A: ";  cin >> A;  cout << "Введите число B: ";  cin >> B;  cout << "Введите количество битов n (в A): ";  cin >> n;  cout << "Введите позицию p (влево от которой установить биты в A): ";  cin >> p;  cout << "Введите количество битов m (в B): ";  cin >> m;  cout << "Введите позицию q (в B): ";  cin >> q;  // вывод начальных значений чисел в двоичном формате  cout << "Начальное значение A в двоичном виде: ";  \_itoa\_s(A, tmp, 2);  cout << tmp << endl;  cout << "Начальное значение B в двоичном виде: ";  \_itoa\_s(B, tmp, 2); // Преобразуем число B в строку в двоичном формате  cout << tmp << endl;  // установка в 1 n битов в A влево от позиции p  int maskA = ((1 << n) - 1) << p; // создаём маску для установки 1  A |= maskA; // устанавливаем биты в 1  // извлечь n битов из A  int bitsFromA = (A & maskA) >> p; // Извлечённые биты  // очищаем m битов в B начиная с позиции q  int maskB = ~(((1 << m) - 1) << q); // Маска для очистки m битов  B = B & maskB;  // заменяем m битов в B на извлечённые биты из A  B = B | (bitsFromA << q);  cout << "Число A в двоичном виде (после установки битов): ";  \_itoa\_s(A, tmp, 2);  cout << tmp << endl;  cout << "Число B в двоичном виде (после изменений): ";  \_itoa\_s(B, tmp, 2);  cout << tmp << endl;  return 0;  } |