**Лабораторная работа № 11. Битовые операции**

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 3. В программе, записанной в правой части, используются различные битовые операции.  Внести изменения в программу с тем, чтобы проверялось число на кратность четырем. | #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_ALL, "RU");  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;  if ((A & 3) == 0) {  cout << "Число кратно 4" << endl;  }  else {  cout << "Число не кратно 4" << endl;  }  } |
| 4. В правой части приведен пример программы, демонстрирующей использование битовых операций.  Проанализировать текст программы и написать пояснения.  5. В соответствии со своим вариантом разработать программы, использующие ***битовые*** операции для решения задач, представленных в таблице.  1) Установить в ноль каждый третий значащий бит целого числа А.  2) Извлечь 3 бита числа **А**, начиная с позиции **n**, и вставить в число **В**, начиная с позиции **m**  **Доп. Задания**  Вариант 5   1. Определить, насколько в числе **А** больше значащих битов, равных единице, чем битов, равных нулю.   Установить в 1 **n** битов в числе **А** вправо от позиции **p**, заменить ими **m** битов числа **В**, начиная с позиции **q**. | **#include <iostream>**  **using namespace std;**  **void main()**  **{**  **setlocale(LC\_CTYPE, "Russian");**  **char tmp[33]; //Буфер для хранения строкового представления числа в двоичном формате.**  **int A, B, maskA = 14; //Число 14 в двоичной системе выглядит как 00001110**  **int maskB = ~maskA >> 1; //Побитовый сдвиг вправо на 1 разряд**  **cout << "Первое число А="; cin >> A;**  **cout << "Второе число В="; cin >> B;**  **\_itoa\_s(A, tmp, 2);//Преобразует число в строковое представление в указанной сc**  **cout << "A=" << tmp << endl;**  **\_itoa\_s(B, tmp, 2);**  **cout << "B=" << tmp << endl;**  **\_itoa\_s(maskA, tmp, 2);//Преобразует maskA в двоичный формат**  **cout << "Маска для А: " << tmp << endl;**  **\_itoa\_s((A & maskA) >> 1, tmp, 2);//оператор И сравнивает 2 результата 00=0 01=0 11=1**  **cout << "Выделенные биты А: " << tmp << endl;**  **\_itoa\_s(maskB, tmp, 2);**  **cout << "Маска для В: " << tmp << endl;**  **\_itoa\_s(B & maskB, tmp, 2);**  **cout << " Очищены биты в B: " << tmp << endl;**  **\_itoa\_s(((B & maskB) | ((A & maskA) >> 1)), tmp, 2);//опиратор ИЛИ 00=0 01=1 11=1**  **cout << " Результат B=" << tmp << endl;**  **}**    **#include <iostream>**  **using namespace std;**  **int main() {**  **setlocale(LC\_CTYPE, "Russian");**  **unsigned int A;**  **cout << "Введите число A: ";**  **cin >> A;**  **unsigned int mask = 0;**  **for (int i = 2; i < 32; i += 3) {**  **mask |= (1 << i);**  **}**  **char tmp[33];**  **\_itoa\_s(A, tmp, 2);**  **cout << "Число A в двоичном виде: " << tmp << endl;**  **unsigned int result = A & ~mask;**  **\_itoa\_s(mask, tmp, 2);**  **cout << "Маска для обнуления: " << tmp << endl;**  **\_itoa\_s(result, tmp, 2);**  **cout << "Результат после обнуления каждого третьего бита: " << tmp << endl;**  **return 0;**  **}**      **#include <iostream>**  **using namespace std;**  **int main() {**  **setlocale(LC\_ALL, "ru");**  **char tmp[33];**  **int A, B, n, m;**  **cout << "Введите число A: "; cin >> A;**  **\_itoa\_s(A, tmp, 2);**  **cout << "A=" << tmp << endl;**  **cout << "Введите число B: "; cin >> B;**  **\_itoa\_s(B, tmp, 2);**  **cout << "B=" << tmp << endl;**  **cout << "Введите начальную позицию в A (n): "; cin >> n;**  **cout << "Введите начальную позицию в B (m): "; cin >> m;**  **unsigned int mask = 7;**  **unsigned int extracted\_bits = (A >> n) & mask;**  **mask = 7 << m;**  **B &= ~mask;**  **unsigned int result = B |= (extracted\_bits << m);**  **cout << "Число B (после вставки): " << result << endl;**  **\_itoa\_s(result, tmp, 2);**  **cout << "Число в двоичной сс: " << tmp << endl;**  **return 0;**  **}**    **#include <iostream>**  **using namespace std;**  **int countBitsDifference(unsigned int A) {**  **int onesCount = 0;**  **int zerosCount = 0;**  **int totalBits = 0;**  **unsigned int temp = A;**  **while (temp > 0) {**  **totalBits++;**  **temp >>= 1;**  **}**  **for (int i = 0; i < totalBits; ++i) {**  **if (A & 1) {**  **onesCount++;**  **}**  **else {**  **zerosCount++;**  **}**  **A >>= 1;**  **}**  **return onesCount - zerosCount;**  **}**  **int main() {**  **setlocale(LC\_ALL, "ru");**  **char tmp[33];**  **unsigned int A;**  **cout << "Введите число A: ";**  **cin >> A;**  **\_itoa\_s(A, tmp, 2);**  **cout << "A=" << tmp << endl;**  **int result = abs(countBitsDifference(A));**  **cout << "Разница между количеством единичных и нулевых битов: " << result << endl;**  **return 0;**  **}**    **unsigned int extractBits(unsigned int A, int p, int n) {**  **unsigned int mask = (1 << n) - 1;**  **return (A >> p) & mask;**  **}**  **// Функция для замены m битов числа B начиная с позиции q на заданные биты**  **unsigned int replaceBits(unsigned int B, unsigned int bits, int q, int m) {**  **unsigned int mask = ((1 << m) - 1) << q;**  **B &= ~mask;**  **B |= (bits << q) & mask;**  **return B;**  **}**  **int main() {**  **setlocale(LC\_ALL, "ru");**  **unsigned int A, B;**  **int p, q, n, m;**  **char tmp[33];**  **cout << "Введите число A: ";**  **cin >> A;**  **\_itoa\_s(A, tmp, 2);**  **cout << "A=" << tmp << endl;**  **cout << "Введите число B: ";**  **cin >> B;**  **\_itoa\_s(B, tmp, 2);**  **cout << "B=" << tmp << endl;**  **cout << "Введите количество битов (n) для извлечения из A: ";**  **cin >> n;**  **cout << "Введите начальную позицию (p) для извлечения битов из A: ";**  **cin >> p;**  **cout << "Введите начальную позицию (q) для замены битов в B: ";**  **cin >> q;**  **cout << "Введите количество битов (m) для замены в B: ";**  **cin >> m;**  **unsigned int extractedBits = extractBits(A, p, n);**    **unsigned int updatedB = replaceBits(B, extractedBits, q, m);**  **cout << "Результат числа B после замены: " << updatedB << endl;**  **\_itoa\_s(updatedB, tmp, 2);**  **cout << "B изменённое=" << tmp << endl;**  **return 0;**  **}** |