**1 Лабораторная работа № 11. Битовые операции**

На языке С/С++ предусмотрены битовые операции для работы с отдельными битами. Их **нельзя** применять к переменным **вещественного** типа. Основные битовые операции:

**− AND (и) − &** **−** если какой-то бит в одном из операндов равен 0, то результирующий бит тоже будет равен 0;

**− OR (или)** **− |** **−** если какой-то бит в одном из операндов равен 1, то результирующий бит тоже будет 1;

**− XOR** (исключающее **или**) **− ^** **−** результирующий бит равен 1, если сравниваемые биты различны;

**− NOT (не)** **− ~ −** меняются все биты на противоположные;

**− сдвиг влево** **− << −** удваивается значение; **−** **сдвиг вправо − >>** – значение уменьшается в два раза.

|  |  |
| --- | --- |
| **Задание** | **Краткие теоретические сведения** |
| 1. Изучить использование битовых операций и маскирования числа, опробовав программу, записанную в правой части, с различными исходными числами. | #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  unsigned int value; int i;  const unsigned int mask = 1 << 31;  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(' ');  }  }    Пример программы, печатающей тридцатидвухразрядное двоичное представление целого числа, введенного с клавиатуры. Здесь используется маскирование всех битов числа, за исключением текущего, выводимого на печать.  Маска имеет вид:  10000000 00000000 00000000 00000000, т.е. 1 << 31.  Последовательно применяется маска и сдвигается число на разряд влево. |
| 2. Выполнить программу, записанную в правой части. Ознакомиться с результатом.  Опробовать программу, изменяя различные биты различных чисел. | **#include <iostream>**  **using namespace std;**  **void main()**  **{**  **setlocale(LC\_CTYPE, "Russian");**  **int A = 150; char tmp[33];**  **\_itoa\_s(A, tmp, 2);**  **cout << " Число А: " << tmp << endl;**  **\_itoa\_s(0x24, tmp, 2);**  **cout << " Маска для А: " << tmp << endl;**  **\_itoa\_s(A | 0x24, tmp, 2);**  **cout << " Результат: " << tmp <<endl<<endl;**  **}**    Пример. Установить в единицу каждый **третий** по счету бит числа А считая справа.  Здесь для вывода двоичного представления числа используется стандартная функция:  **\_itoa\_s (**число ввода, строка вывода, основание системы счисления**)**. |
| 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\_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;**  **}**  #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)  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. | **#include <iostream>**  **using namespace std;**  **void main()**  **{**  **setlocale(LC\_CTYPE, "Russian");**  **char tmp[33];**  **int A, B, maskA = 14;**  **int maskB = ~maskA >> 1;**  **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);**  **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);**  **cout << " Результат B=" << tmp << endl;**  **}**  Сначала вводятся два числа, потом образуются маски для них. Маска(Б) это инвертированная маска(А сдвинутая на 1 влево. Выводится двоичное представление всех чисел. Потом биты очищаются с помощью масок  Пример. Извлечь 3 бита числа А, начиная со второго по счету справа и вставить их в число В, начиная с первого по счету справа. |

5. В соответствии со своим вариантом разработать программы, использующие ***битовые*** операции для решения задач, представленных в таблице.

Результаты одной из программ представить в **Отладчике**.

.

| **№ варианта** | **Условия задач** |
| --- | --- |
| **11** | 1. Используя битовые операции проверить, кратно ли шестнадцати число **А**.   #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "ru");  int a, b;  cout << "Введите число "; cin >> a;  b = a;  a = a >> 4;  a = a << 4;  if (b == a) {  cout << "Кратно 16" << endl;  }  else {  cout << "Не кратно 16 " << endl;  }  }     1. Установить в 0 **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, m, q;  int sum = 0;  cout << "Введите число "; cin >> b;  cout << "Введите сдвиг влево от позиции: "; cin >> q;  cout << "m = "; cin >> m;  for (int i = 0; i < q; i++) {  sum += pow(2, i);  }  a = b;  a = a & sum;  b = b >> q + m;  b = b << q + m;  b = b | a;  cout << "Конечное число: " << b;  } |

|  |  |
| --- | --- |
| **8** | 1. Установить в ноль каждый третий значащий бит целого числа А.   #include <iostream>  #include <bitset>  using namespace std;  int main() {  setlocale(LC\_ALL, "RU");  int a, b;  cout << "Введите число "; cin >> a;  int mask = 0;  for (int i = 2; i < sizeof(int) \* 8; i+=3) {  mask |= (1U << i);  }  cout << "Исходное число " << a << " (" << bitset<10>(a) << ")" << endl;  a &= ~mask;  cout << "Конечное число " << a << " (" << bitset<10>(a) << ")" << endl;  }     1. Извлечь 3 бита числа **А**, начиная с позиции **n**, и вставить в число **В**, начиная с позиции **m**.   #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "RU");  int a, b, m, n = 3;  int sum = 0;  cout << "Введите число "; cin >> b;  cout << "С позиции: "; cin >> m;  for (int i = 0; i < m; i++) {  sum += pow(2, i);  }  a = b;  a = a & sum;  b = b >> m + n;  b = b << m + n;  b = b | a;  cout << "Конечное число: " << b;  } |

|  |  |
| --- | --- |
| **2** | 1. Извлечь 5 битов числа **A**, начиная со второго и вставить их в число **B**, начиная с третьего бита.   #include <iostream>  using namespace std;  int main() {  setlocale(LC\_ALL, "RU");  unsigned int A, B;  cout << "Введите число A: ";  cin >> A;  cout << "Введите число B: ";  cin >> B;    unsigned int extractedBits = (A >> 1) & 0x1F;    B = (B & ~(0x1F << 2)) | (extractedBits << 2);  std::cout << "Результат B: " << B << std::endl;  return 0;  }     1. Установить в 1в числе **А n** битов вправо от позиции **p**.     #include <iostream>  int main() {  setlocale(LC\_ALL, "RU");  unsigned int A;  int p, n;  std::cout << "Введите число A: ";  std::cin >> A;  std::cout << "Введите позицию p: ";  std::cin >> p;  std::cout << "Введите количество битов n: ";  std::cin >> n;    unsigned int mask = ((1U << n) - 1) << p;  A |= mask;  std::cout << "Результат A: " << A << std::endl;  return 0;  } |

|  |  |
| --- | --- |
| 6 | 1. Установить в единицу каждый второй значащий бит целого числа **А**.   #include <iostream>  unsigned int setEverySecondSignificantBit(unsigned int A) {  unsigned int result = A;  unsigned int count = 0;    for (int i = 0; i < 32; ++i) {    if (A & (1U << i)) {  count++; //  if (count % 2 == 0) {  result |= (1U << i);  }  }  }  return result;  }  int main() {  unsigned int A;  setlocale(LC\_ALL, "RU");  std::cout << "Введите число A: ";  std::cin >> A;  unsigned int result = setEverySecondSignificantBit(A);  std::cout << "Результат: " << result << std::endl;  return 0;  }     1. Извлечь 3 бита числа **А**, начиная с позиции **n**, и вставить в число **В**, начиная с позиции **m**.     // Извлечь 3 бита числа А, начиная с позиции n, и вставить в число В, начиная с позиции m.  #include <iostream>  using namespace std;  unsigned int extractBits(unsigned int A, int n) {  return (A >> n) & 7;  }  unsigned int insertBits(unsigned int B, unsigned int extractedBits, int m) {    unsigned int mask = ~(7 << m);  B &= mask;    B |= (extractedBits << m);  return B;  }  int main() {  unsigned int A, B;  int n, m;  setlocale(LC\_ALL, "RU");    cout << "Введите число A: ";  cin >> A;  cout << "Введите число B: ";  cin >> B;  cout << "Введите позицию n (для извлечения 3 бит из A): ";  cin >> n;  cout << "Введите позицию m (для вставки в B): ";  cin >> m;    unsigned int extractedBits = extractBits(A, n);    B = insertBits(B, extractedBits, m);  cout << "Результат: " << B << endl;  return 0;  } |