Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 11

По дисциплине «Основы алгоритмизации и программирования»

На тему «Битовые операции»

Выполнил:

Студент 1 курса 6 группы

Кучерук Николай Петрович

Преподаватель: асс. Андронова М.В.

2023, Минск

**Задание**

| **№ варианта** | **Условия задач** |
| --- | --- |
| **9** | 1. Извлечь 5 битов числа **A**, начиная с третьего по счету справа, и вставить их в число **B**, начиная со 2 по счету справа. 2. Установить в 0 **n** битов в числе **А** вправо от позиции **p**. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  char tmp[33];//объявляем массив символов для вывода двоичного представления чисел,размер выбран по принципу того, что тип данных int вмещает в себя 32 бит  int A, B, maskA = 124;//т.к. 124 в 2 представлении 11111000, то при использовании & мы найдем какие биты расположены в нужных нам позициях  int maskB = ~maskA >> 1;//чтобы очисть место для постановки преобразовываем маскуА в ей обратную 1...1100000111 и при помощи операции NOT и сдвигаем вправо на 1,т.к. очищать нужно со 2 бита,а маскаА начинается с 3  cout << "Первое число А="; cin >> A;  cout << "Второе число В="; cin >> B;  \_itoa\_s(A, tmp, 2);//используем функцию \_itoa\_s для преобразования данных типа int в string(что представляет из себя массив символов, который мы объявили выше)  cout << "A=" << tmp << endl;//выводим двоичное представление A  \_itoa\_s(B, tmp, 2);  cout << "B=" << tmp << endl;//выводим двоичное представление B  \_itoa\_s((A & maskA) >> 2, tmp, 2);//при операции & с маской остаются только те биты,которые попадают в диапазон маски, смещаем на 2,т.к. выделяли начиная с 3  cout << "Выделенные биты А: " << tmp << endl;//выводим те биты, которые мы извлекли из числа A  \_itoa\_s(B & maskB, tmp, 2);//т.к. мы преобразовали маску операций NOT и сместили на нужный нам диапазон, то все биты, которые попали в диапазон данной маски, будут очищены  cout << " Очищены биты в B: " << tmp << endl;  \_itoa\_s(((B & maskB) | ((A & maskA) >> 1)), tmp, 2);//при операции OR с очищенным B и выделенными битами(смещаем вправо на 1,т.к. брали начиная с 3,а объединяем со 2) единицы, которые есть в выделенных битах, встанут на места очищенных нулей  cout << " Результат B=" << tmp << endl;  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  char tmp[33];  int A, B, n, p;  cout << "Число А="; cin >> A;  \_itoa\_s(A, tmp, 2);  cout << "A=" << tmp << endl;  cout << "Какое число битов вы хотите установить в 0? :"; cin >> n;  cout << "Начиная с какого элемента вправо? :"; cin >> p;  p = p - n;//чтобы мы могли установить 0 вправо от позиции, нужно установить такую позицию, чтобы индекс самого большого разряда нашей маски находился на введённой позиции  n = pow(2, n) - 1;//каждая степень двойки в двоичном представлении имеет вид 1..0,где кол-во 0 зависит напрямую от степени числа 2,чтобы все эти нули были заполнены 1,надо взять предшествующее число  p > 0 ? n = n << p : n = n >> abs(p); //сдвигаем вправо на нужную позицию  int mask = ~n;//инвертируем маску, чтобы в ее диапазоне мы могли устанавливать биты в 0  \_itoa\_s(mask, tmp, 2);  cout << "Маска для А: " << tmp << endl;//выведем маску для возможности её анализа  \_itoa\_s(A & mask, tmp, 2);//при операции & и инвертированной маске, все числа в её диапазоне переходят в 0,а вне сохраняют свое изначальное значение  cout << " Результат A: " << tmp << endl;  } |
| **Результат программы** |
|  |

Дополнительные варианты

| **№ варианта** | **Условия задач** |
| --- | --- |
| **4** | 1. Используя битовые операции проверить, кратно ли четырем число **А**. 2. Установить в 1 в числе **А n** битов влево от позиции **p**. |

|  |
| --- |
| **Код программы** |
| #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)//числа кратные 4 в двоичном представлении имеют вид 1....00,поэтому если при операции & с 3(11в двоичном представлении) будет хоть какой-то результат не равный 0,то значит там есть биты 1, следовательно кратным не будет  cout << "Число кратно 4" << endl;  else  cout << "Число не кратно 4" << endl;  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  char tmp[33];  int A, B, n, p;  cout << "Число А="; cin >> A;  \_itoa\_s(A, tmp, 2);  cout << "A=" << tmp << endl;  cout << "Какое число битов вы хотите установить в 1? :"; cin >> n;  cout << "Начиная с какого элемента? :"; cin >> p;  n = pow(2, n) - 1;//каждая степень двойки в двоичном представлении имеет вид 1..0,где кол-во 0 зависит напрямую от степени числа 2,чтобы все эти нули были заполнены 1,надо взять предшествующее число  n = n << p-1;//сдвигаем вправо на желаемую позицию. -1,чтобы позиция была включительно  int mask = n;  \_itoa\_s(mask, tmp, 2);  cout << "Маска для А: " << tmp << endl;//выведем маску для возможности её анализа  \_itoa\_s(A | mask, tmp, 2);//при операции | и инвертированной маске, все числа в её диапазоне переходят в 1,а вне сохраняют свое изначальное значение  cout << " Результат A: " << tmp << endl;  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условия задач** |
| --- | --- |
| **2** | 1. Извлечь 5 битов числа **A**, начиная со второго и вставить их в число **B**, начиная с третьего бита. 2. Установить в 1в числе **А n** битов вправо от позиции **p**. |

|  |
| --- |
| **Код программы** |
| #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  char tmp[33];//объявляем массив символов для вывода двоичного представления чисел,размер выбран по принципу того, что тип данных int вмещает в себя 32 бит  int A, B, maskA = 62;//т.к. 62 в 2 представлении 111110, то при использовании & мы найдем какие биты расположены в нужных нам позициях  int maskB = ~(maskA << 1) ;//чтобы очисть место для постановки преобразовываем маскуА в ей обратную 1...1110000011 и при помощи операции NOT и сдвигаем влево на 1,т.к. очищать нужно со 3 бита, а маскаА начинается с 2  //скобки нужны для того, чтобы когда мы делали сдвиг влево,0 который от него остается тоже инвертировался при операции ~  cout << "Первое число А="; cin >> A;  cout << "Второе число В="; cin >> B;  \_itoa\_s(A, tmp, 2);//используем функцию \_itoa\_s для преобразования данных типа int в string(что представляет из себя массив символов, который мы объявили выше)  cout << "A=" << tmp << endl;//выводим двоичное представление A  \_itoa\_s(B, tmp, 2);  cout << "B=" << tmp << endl;//выводим двоичное представление B  \_itoa\_s((A & maskA) >> 1, tmp, 2);//при операции & с маской остаются только те биты,которые попадают в диапазон маски, смещаем на 1,т.к. выделяли начиная с  cout << "Выделенные биты А: " << tmp << endl;//выводим те биты, которые мы извлекли из числа A  \_itoa\_s(B & maskB, tmp, 2);//т.к. мы преобразовали маску операций NOT и сместили на нужный нам диапазон, то все биты, которые попали в диапазон данной маски, будут очищены  cout << " Очищены биты в B: " << tmp << endl;  \_itoa\_s(((B & maskB) | ((A & maskA) << 1)), tmp, 2);//при операции OR с очищенным B и выделенными битами(смещаем влево на 1,т.к. брали начиная с 2,а объединяем со 3) единицы, которые есть в выделенных битах, встанут на места очищенных нулей  cout << " Результат B=" << tmp << endl;  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  char tmp[33];  int A, B, n, p;  cout << "Число А="; cin >> A;  \_itoa\_s(A, tmp, 2);  cout << "A=" << tmp << endl;  cout << "Какое число битов вы хотите установить в 1? :"; cin >> n;  cout << "Начиная с какого элемента вправо? :"; cin >> p;  p = p - n;//чтобы мы могли установить 0 вправо от позиции, нужно установить такую позицию, чтобы индекс самого большого разряда нашей маски находился на введённой позиции  n = pow(2, n) - 1;//каждая степень двойки в двоичном представлении имеет вид 1..0,где кол-во 0 зависит напрямую от степени числа 2,чтобы все эти нули были заполнены 1,надо взять предшествующее число  p > 0 ? n = n << p : n = n >> abs(p); //сдвигаем вправо на нужную позицию  int mask =n;  \_itoa\_s(mask, tmp, 2);  cout << "Маска для А: " << tmp << endl;//выведем маску для возможности её анализа  \_itoa\_s(A | mask, tmp, 2);//при операции | и инвертированной маске, все числа в её диапазоне переходят в 1,а вне сохраняют свое изначальное значение  cout << " Результат A: " << tmp << endl;  } |
| **Результат программы** |
|  |

| **№ варианта** | **Условия задач** |
| --- | --- |
| **15** | 1. Используя битовые операции проверить, кратно ли двум число **А**. 2. Установить в 0 **n** битов в числе **А** влево от позиции **p**. |

|  |
| --- |
| **Код программы** |
| #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 & 1) == 0)//числа кратные 2 в двоичном представлении имеют вид 1....0,поэтому если при операции & с 1(1 в двоичном представлении) будет хоть какой-то результат не равный 0,то значит там есть бит 1,следовательно кратным не будет  cout << "Число кратно 2" << endl;  else  cout << "Число не кратно 2" << endl;  } |
| **Результат программы** |
|  |

|  |
| --- |
| **Код программы** |
| #include <iostream>  using namespace std;  void main()  {  setlocale(LC\_CTYPE, "Russian");  char tmp[33];  int A, B, n, p;  cout << "Число А="; cin >> A;  \_itoa\_s(A, tmp, 2);  cout << "A=" << tmp << endl;  cout << "Какое число битов вы хотите установить в 0? :"; cin >> n;  cout << "Начиная с какого элемента? :"; cin >> p;  n = pow(2, n) - 1;//каждая степень двойки в двоичном представлении имеет вид 1..0,где кол-во 0 зависит напрямую от степени числа 2,чтобы все эти нули были заполнены 1,надо взять предшествующее число  n = n << p - 1;//сдвигаем вправо на желаемую позицию. -1,чтобы позиция была включительно  int mask = ~n;//инвертируем маску, чтобы в ее диапазоне мы могли устанавливать биты в 0  \_itoa\_s(mask, tmp, 2);  cout << "Маска для А: " << tmp << endl;//выведем маску для возможности её анализа  \_itoa\_s(A & mask, tmp, 2);//при операции & и инвертированной маске, все числа в её диапазоне переходят в 0,а вне сохраняют свое изначальное значение  cout << " Результат A: " << tmp << endl;  } |
| **Результат программы** |
|  |