|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра Математического обеспечения и стандартизации информационных технологий | |
|  | |
|  | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1** | |
| **по дисциплине** | |
| **«**Структуры и алгоритмы обработки данных**»**  **Тема: «Поразрядные операции и их применение»** | |
|  | |
| Выполнил студент группы ИКБО-06-21 | Школьник Т.О |
| Принял преподаватель | Филатов А.С. |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись руководителя)* |

Москва 2022

# **Цель работы**

Получение навыков применения поразрядных операций в алгоритмах.

# **Постановка задачи**

1. Разработать программу, которая продемонстрирует выполнение упражнений варианта. Результаты выполнения упражнения выводить на монитор.

Требования к упражнениям:

1. Определить переменную целого типа, присвоить ей значение, используя константу в шестнадцатеричной системе счисления. Разработать функцию, которое установит заданные в задании биты исходного значения переменной в значение 1, используя соответствующую маску и поразрядную операцию.
2. Разработать функцию, которая обнуляет заданные в задании биты исходного значения целочисленной переменной, введенной пользователем, используя соответствующую маску и поразрядную операцию.
3. Разработать функцию, которая умножает значение целочисленной переменной, введенной пользователем, на множитель, используя соответствующую поразрядную операцию.
4. Разработать функцию, которая делит значение целочисленной переменной, введенной пользователем, на делитель, используя соответствующую поразрядную операцию.
5. Разработать функцию, реализующую задание, в которой используются только поразрядные операции. В выражении используется маска – переменная. Маска инициализируется единицей в младшем разряде (маска 1) или единицей в старшем разряде (маска 2). Изменяемое число и n вводится с клавиатуры.
6. Провести тестирование программы на небольших объемах данных, введенных вручную. Разработанные тесты должны покрывать все случаи входных данных (средний, лучший, худший). Результаты тестирования свести в сводные таблицы.
7. Составить отчет, отобразив в нем описание выполнения всех этапов разработки, тестирования и код всей программы со скриншотами результатов тестирования.

Вариант №9. Условие задания:

|  |  |
| --- | --- |
| Упражнение 1 | 0-ый, 11-ый, 3-ий |
| Упражнение 2 | Четыре старших |
| Упражнение 3 | 4 |
| Упражнение 4 | 4 |
| Упражнение 5 | Обнулить n-ый бит, используя маску 2 |

# **Решение**

Поразрядные операции требуются для изменений отдельных битов в значениях. Это широко используется:

1. в различных быстрых алгоритмах сортировок;
2. оптимизации часто используемых операций (например, приведения к верхнему или нижнему регистру, обмен значения переменных);
3. для более оптимального использования кеш-памяти;
4. при работе с IP-адресами.

В языке программирования С++ существует шесть основных поразрядных операций:

1. Сдвиг влево двоичного кода (x << n), используется для умножения текущего числа на 2n;
2. Сдвиг вправо двоичного кода (x >> n), используется для деления текущего числа на 2n;
3. Поразрядное И (x & mask), используется для записи 0 в требуемый разряд текущего числа;
4. Поразрядное ИЛИ (x | mask), используется для записи 1 в требуемый разряд текущего числа;
5. Исключающее ИЛИ (x ^ mask), используется для проверки битов двух переменных;
6. Инверсия (~x), используется для инверсии определенных битов переменной.

Упражнение 1

Требуется с помощью маски установить 0, 11, 3 биты в единицу. С помощью применения операции поразрядного ИЛИ между маской, которой соответствует число 0x0809 в 16 С.С. и заданным пользователем числа происходит установка определенных битов в требуемое значение.

|  |
| --- |
| //Функция установки битов исходного значения переменной в значение 1 (упр. 1)  void setBitsToOne(unsigned int& value)  {  value |= 0x0809;  } |

Упражнение 2

Требуется с помощью маски установить 4 старших бита в ноль. С помощью применения операции поразрядного И между маской, которой соответствует число 0x0FFF в 16 С.С. и заданным пользователем числа происходит установка определенных битов в требуемое значение.

|  |
| --- |
| //Функция установки битов исходного значения переменной в значение 0 (упр. 2)  void setBitsToZero(unsigned int& value)  {  value &= 0x0FFF;  } |

Упражнение 3

Требуется умножить заданное число на 4. С помощью применения операции сдвига двоичного кода влево на значение, равное 2 происходит умножение значения переменной на 4.

|  |
| --- |
| //Функция умножения значения целочисленной переменной (упр. 3)  void multiplyValue(unsigned int& value)  {  value <<= 2;  } |

Упражнение 4

Требуется разделить заданное число на 4. С помощью применения операции сдвига двоичного кода вправо на значение, равное 2 происходит деление значения переменной на 4.

|  |
| --- |
| //Функция деления значения целочисленной переменной (упр. 4)  void divideValue(unsigned int& value)  {  value >>= 2;  } |

Упражнение 5

Требуется установить заданный пользователем бит в ноль используя маску 0010. С помощью применения операции сдвига двоичного кода вправо на 4, переставляем 1 в младший бит маски. Далее с помощью применения операции сдвига двоичного кода влево, на заданное пользователем значение, над маской, младший бит которой теперь установлен в 1, получается маска, где единица стоит на номере заданного пользователем бита, а остальные биты – нули. Применяя инверсию к маске, а после операцию поразрядного И между маской и значением переменной конкретный бит заданного значения переменной будет установлен в 0.

|  |
| --- |
| //Функция установки определенных битов значения переменной в 0 (упр. 5)  void setBitToZero(unsigned int& value, unsigned int numOfBit)  {  unsigned int mask = 0x0010;  value &= (~((mask >> 4) << numOfBit));  } |

В программе имеется два режима работы, тестовый прогон, данные которого вводятся вручную, и рабочий прогон, в которых происходит автоматическое тестирование. Все, что требуется, это выбрать режим работы и ввести требуемую информацию.

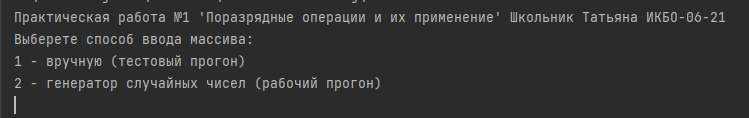


Рисунок 1. Интерфейс программы

# **Тестирование**

В тестовом прогоне выберем пять чисел, пусть это будут: 17, 6003, 65036, 1, 0. Для упражнения 5 выберем биты под номерами: 0, 12, 3, 15, 6.

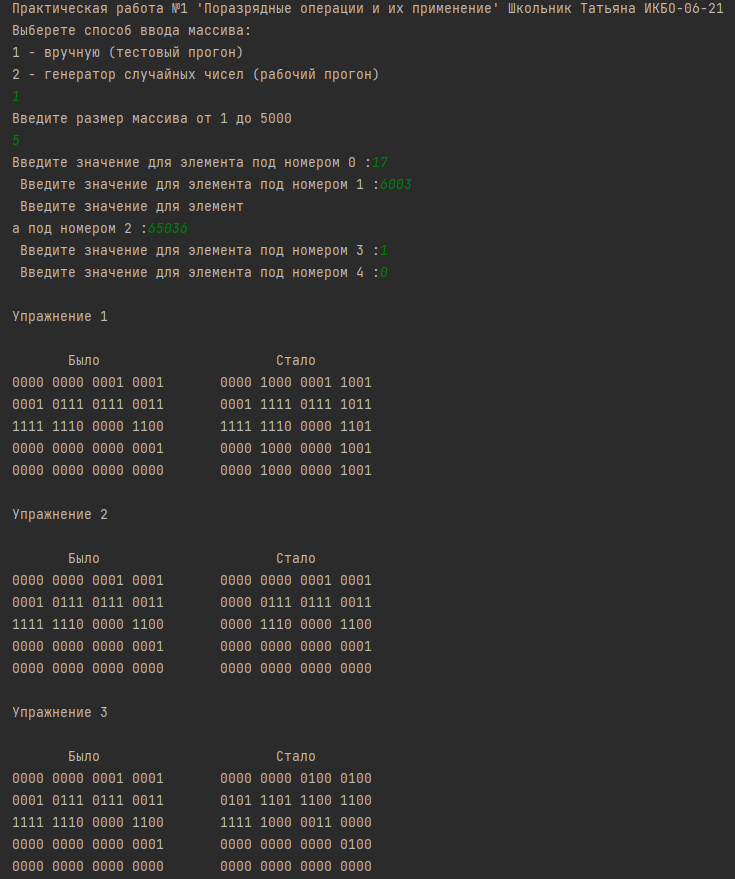


Рисунок 2. Тестирование программы

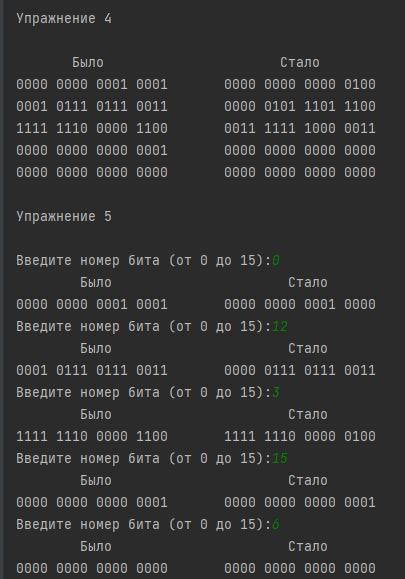


Рисунок 3. Тестирование программы

Рабочий прогон программы начинается с установки кол-ва элементов исходного массива, после чего для удобства оперирования данными, исходный массив копируется в рабочий массив, над которыми будут произведена работа при помощи функций. Рабочий прогон покрывает положительные числа в интервале от 0 до 65535. Производится на объемах из 1000, 2000, 3000, 4000, 5000 элементов.

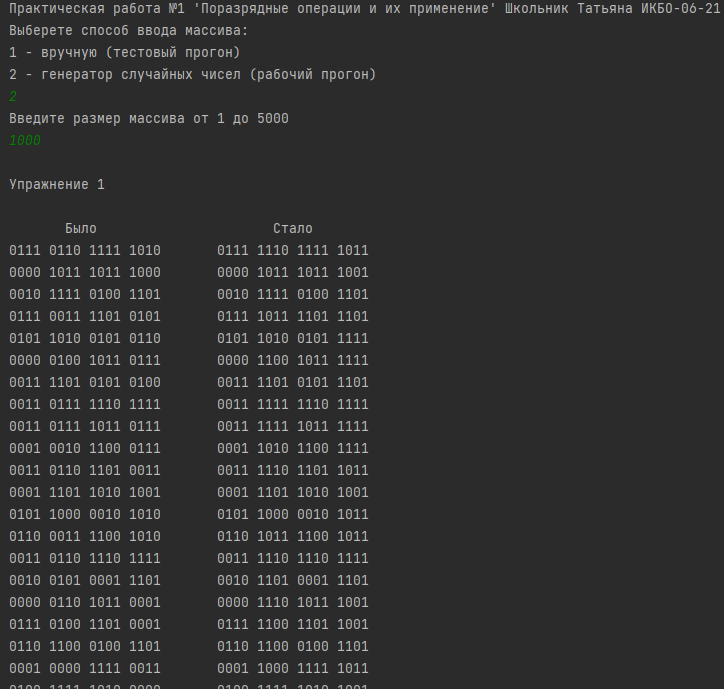


Рисунок 4. Тестирование программы

# **Вывод**

В результате выполнения работы я:

1. Освоила алгоритмы работы с поразрядными операциями и их реализацию на языке программирования C++
2. Научилась программировать автоматическое тестирование простых программ

# **Исходный код программы**

|  |
| --- |
| //  // Created by t.shkolnik  //  #include <iostream>  #include <time.h>  #include <windows.h>  using namespace std;  //Функция побитового вывода значения переменной  void printValue(const unsigned int& value)  {  int numByte = sizeof(int) \* 4;  unsigned mask = (1 << (numByte - 1));  for (int i = 1; i <= numByte; i++)  {  cout << ((value & mask) >> (numByte - i));  mask >>= 1;  if (i % 4 == 0) { cout << " "; }  }  }  //Функция установки битов исходного значения переменной в значение 1 (упр. 1)  void setBitsToOne(unsigned int& value)  {  value |= 0x0809;  }  //Функция установки битов исходного значения переменной в значение 0 (упр. 2)  void setBitsToZero(unsigned int& value)  {  value &= 0x0FFF;  }  //Функция умножения значения целочисленной переменной (упр. 3)  void multiplyValue(unsigned int& value)  {  value <<= 2;  }  //Функция деления значения целочисленной переменной (упр. 4)  void divideValue(unsigned int& value)  {  value >>= 2;  }  //Функция установки определенных битов значения переменной в 0 (упр. 5)  void setBitToZero(unsigned int& value, unsigned int numOfBit)  {  unsigned int mask = 0x0010;  value &= (~((mask >> 4) << numOfBit));  }  int main(void)  {  SetConsoleOutputCP(CP\_UTF8);  srand(time(NULL));  unsigned int\* values;  unsigned int\* valuesTest;  int varietyFlag;  int arraySize;  string temp = " Было Стало \n";  cout << "Практическая работа №1 'Поразрядные операции и их применение' Школьник Татьяна ИКБО-06-21" << endl;  cout << "Выберете способ ввода массива:" << endl;  cout << "1 - вручную (тестовый прогон)" << endl;  cout << "2 - генератор случайных чисел (рабочий прогон)" << endl;  cin >> varietyFlag;  cout << "Введите размер массива от 1 до 5000" << endl;  cin >> arraySize;  values = new unsigned int[arraySize];  valuesTest = new unsigned int[arraySize];  if (varietyFlag == 1)  {  for (int i = 0; i < arraySize; i++)  {  cout << "Введите значение для элемента под номером " << i << " : ";  cin >> values[i];  valuesTest[i] = values[i];  }  }  else if (varietyFlag == 2)  {  for (int i = 0; i < arraySize; i++)  {  values[i] = rand() % 65536;  valuesTest[i] = values[i];  }  }  else { exit(0); }  cout << "\nУпражнение 1\n" << endl;  cout << temp;  for (int i = 0; i < arraySize; i++)  {  printValue(values[i]); cout << " ";  setBitsToOne(valuesTest[i]);  printValue(valuesTest[i]); cout << endl;  valuesTest[i] = values[i];  }  cout << "\nУпражнение 2\n" << endl;  cout << temp;  for (int i = 0; i < arraySize; i++)  {  printValue(values[i]); cout << " ";  setBitsToZero(valuesTest[i]);  printValue(valuesTest[i]); cout << endl;  valuesTest[i] = values[i];  }  cout << "\nУпражнение 3\n" << endl;  cout << temp;  for (int i = 0; i < arraySize; i++)  {  printValue(values[i]); cout << " ";  multiplyValue(valuesTest[i]);  printValue(valuesTest[i]);  valuesTest[i] = values[i]; cout << endl;  }  cout << "\nУпражнение 4\n" << endl;  cout << temp;  for (int i = 0; i < arraySize; i++)  {  printValue(values[i]); cout << " ";  divideValue(valuesTest[i]);  printValue(valuesTest[i]); cout << endl;  valuesTest[i] = values[i];  }  cout << "\nУпражнение 5\n" << endl;  if (varietyFlag == 2)  {  cout << temp;  }  for (int i = 0; i < arraySize; i++)  {  if (varietyFlag == 1)  {  int numBit;  cout << "Введите номер бита (от 0 до 15): ";  cin >> numBit;  cout << temp;  setBitToZero(valuesTest[i], numBit);  }  else  {  setBitToZero(valuesTest[i], rand() % 16);  }  printValue(values[i]); cout << " ";  printValue(valuesTest[i]); cout << endl;  valuesTest[i] = values[i];  }  system("pause"); return 0;  } |