|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | 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**

Побитовые операции выполняются над отдельными разрядами или битами чисел. Данные операции производятся только над целыми числами.

Существует шесть основных поразрядных операций в языке программирования С++:

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

Упражнение 1

Нужно установить 0, 11, 3 биты в единицу. Применяем поразрядное ИЛИ к заданному числу и маске (число 0x0809 в 16 С.С), где единицы стоят на местах битов, которые нужно поменять.

|  |
| --- |
| //Функция устанавливает 0, 3 и 11ый биты переданного значения в 1  void set1InBits(unsigned int& userValue)  {  unsigned int mask = 0x0809; // 0000 1000 0000 1001  userValue |= mask;  } |

Упражнение 2

Нужно установить 4 старших бита в ноль. Применяем поразрядное И к заданному числу и маске (число 0x0FFF в 16 С.С), где единицы стоят на местах битов, которые НЕ нужно менять.

|  |
| --- |
| //Функция устанавливает 4 старших бита переданного значения в 0  void set0ToBits(unsigned int& userValue)  {  unsigned int mask = 0x0FFF; // 0000 1111 1111 1111  userValue &= mask;  } |

Упражнение 3

Требуется умножить заданное число на 4. Нужно применить побитовый сдвиг влево на 2, так как умножение числа происходит на на 2n.

|  |
| --- |
| //Функция умножает значение на 4, используя побитовый сдвиг влево  void multiply(unsigned int& userValue)  {  userValue <<= 2;  } |

Упражнение 4

Требуется разделить заданное число на 4. Нужно применить побитовый сдвиг вправо на 2, так как деление числа происходит на на 2n.

|  |
| --- |
| //Функция делит значение на 4, используя побитовый сдвиг вправо  void divide(unsigned int& userValue)  {  userValue >>= 2;  } |

Упражнение 5

Нужно установить заданный пользователем бит в ноль используя маску 0010 (2). Применяем побитовый сдвиг маски вправо на 4, таким образом переставляем 1 в младший бит маски. Далее применяем побитовый сдвиг 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. Теперь применяя инверсию к маске, а после операцию поразрядного И между маской и значением переменной конкретный бит заданного значения переменной будет установлен в 0.

|  |
| --- |
| //Функция обнуляет определенные биты переданного значения  void setZeros(unsigned int& userValue, unsigned int n)  {  unsigned int mask = 0x0010;  unsigned int maskWith1InEnd = mask >> 4;  unsigned int maskWith1AtNPosition = maskWith1InEnd << n;  userValue &= ~maskWith1AtNPosition;  } |
|  |

|  |
| --- |
| //Функция вывода значения по битам  void printBinary(unsigned int &x) {  int n = sizeof(int) \* 4;  unsigned maska = (1 << (n - 1));  for (int i = 1; i <= n; i++) {  cout << ((x & maska) >> (n - i));  maska >>= 1;  if (i % 4 == 0) { cout << " "; }  }  } |
|  |

|  |
| --- |
| //Вводная функция к каждому заданию, выбор режима работы  void onboading(int taskNum, int \*arraySize) {  cout << "\nУпражнение " << taskNum << "\n" << endl;  cout << "Введите размер массива от 1 до 5000" << endl;  cin >> \*arraySize;  cout << "Выберете способ ввода массива:" << endl;  cout << "1 - вручную (тестовый прогон)" << endl;  cout << "2 - генератор случайных чисел (рабочий прогон)" << endl;  } |

|  |
| --- |
| //Функция считывает элементы массива, введённые пользователем  void readArray(unsigned int \*userValues, unsigned int \*valuesTest, int arraySize) {  for (int i = 0; i < arraySize; i++) {  cout << "Введите значение для элемента " << i << " : ";  cin >> userValues[i];  valuesTest[i] = userValues[i];  }  } |

|  |
| --- |
| //Функция заполняет массив случайными значениями  void generateArray(unsigned int \*userValues, unsigned int \*valuesTest, int arraySize) {  for (int i = 0; i < arraySize; i++) {  userValues[i] = rand() % 65536;  valuesTest[i] = userValues[i];  }  } |

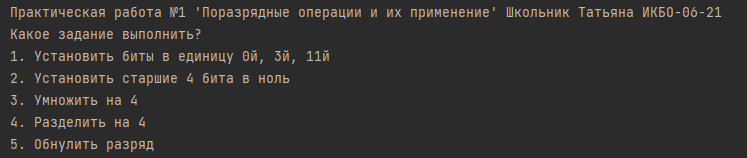
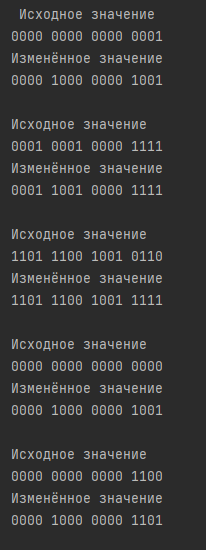
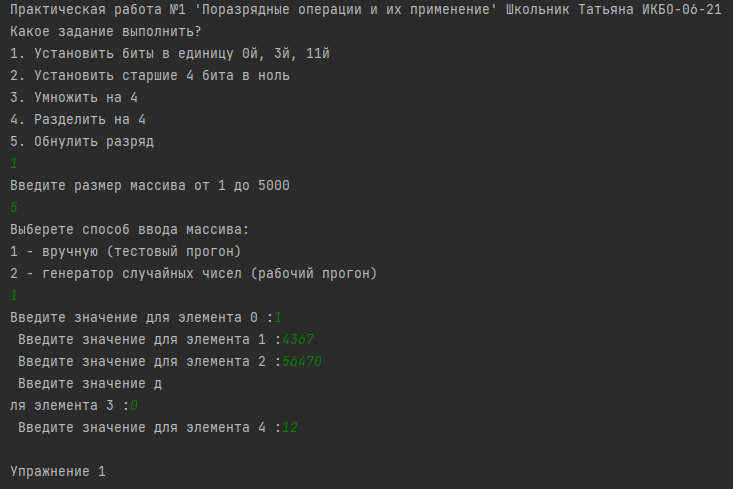
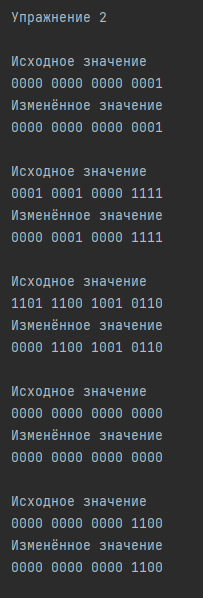
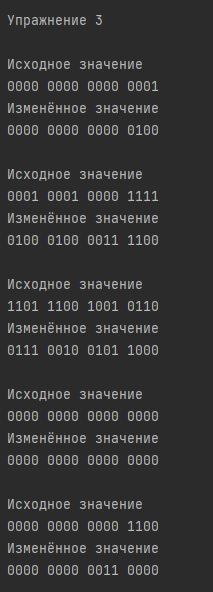
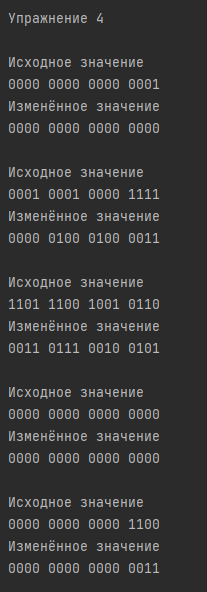


Рисунок 1. Интерфейс программы

# **Тестирование**

В тестовом прогоне выберем пять чисел, пусть это будут: 1, 4367, 56470, 0, 12. Для упражнения 5 выберем биты под номерами: 0, 15, 4, 11, 8.



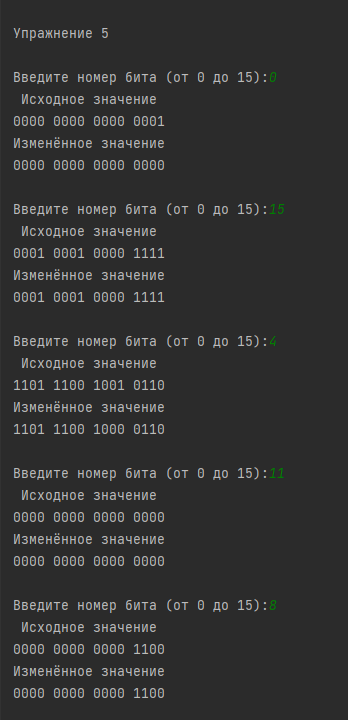


Рисунок 2. Тестирование программы

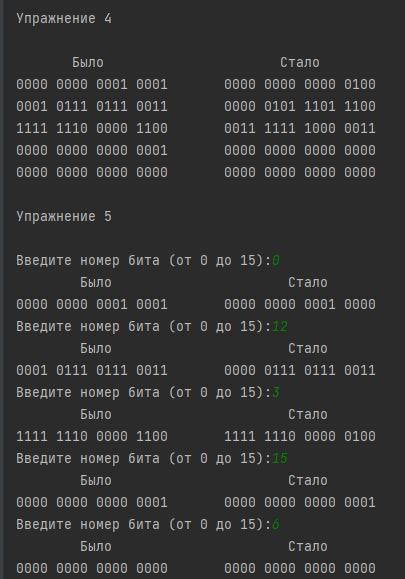


Рисунок 3. Тестирование программы

При рабочем прогоне программы так же устанавливается размер массива и способ ввода. Для удобства оперирования данными, исходный массив копируется в рабочий массив, над которыми будут произведена работа при помощи функций. В процессе прогона покрываются положительные числа в интервале от 0 до 65535.

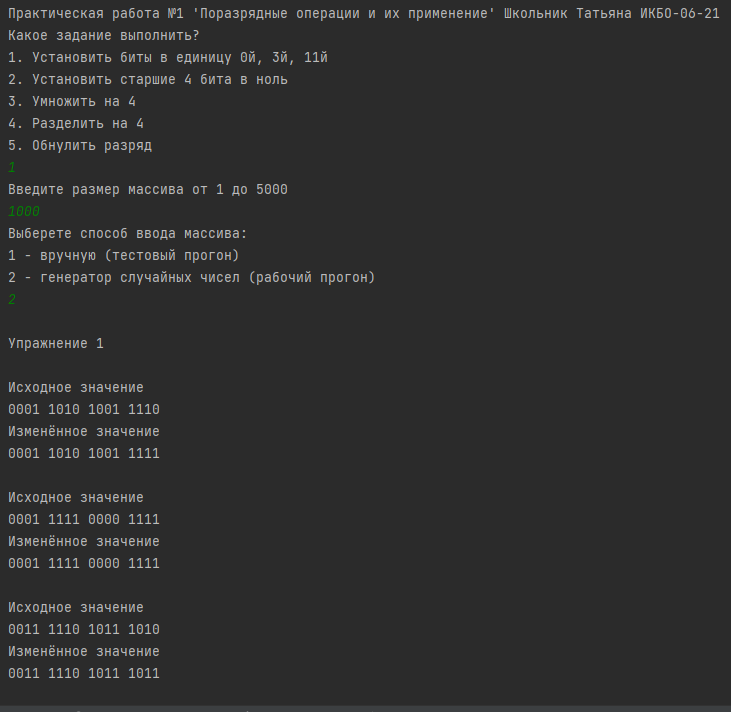


Рисунок 4. Тестирование программы

# **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 вывода значения по битам  void printBinary(unsigned int &x) {  int n = sizeof(int) \* 4;  unsigned maska = (1 << (n - 1));  for (int i = 1; i <= n; i++) {  cout << ((x & maska) >> (n - i));  maska >>= 1;  if (i % 4 == 0) { cout << " "; }  }  }  //Функция устанавливает 0, 3 и 11ый биты переданного значения в 1  void set1InBits(unsigned int &userValue) {  unsigned int maska = 0x0809; // 0000 1000 0000 1001  userValue |= maska;  }  //Функция устанавливает старшие 4 бита переданного значения в 0  void set0ToBits(unsigned int &userValue) {  unsigned int maska = 0x0FFF; // 0000 1111 1111 1111  userValue &= maska;  }  //Функция умножает значение на 4, используя побитовый сдвиг влево  void multiply(unsigned int &userValue) {  userValue <<= 2;  }  //Функция делит значение на 4, используя побитовый сдвиг вправо  void divide(unsigned int &userValue) {  userValue >>= 2;  }  //Функция обнуляет определенные биты переданного значения  void setZeros(unsigned int &userValue, unsigned int n) {  unsigned int maska = 0x0010;  unsigned int maskaWith1InEnd = maska >> 4;  unsigned int maskaWith1AtNPosition = maskaWith1InEnd << n;  userValue &= ~maskaWith1AtNPosition;  }  //Вводная функция к каждому заданию, выбор режима работы  void onboading(int \*arraySize) {  cout << "Введите размер массива от 1 до 5000" << endl;  cin >> \*arraySize;  cout << "Выберете способ ввода массива:" << endl;  cout << "1 - вручную (тестовый прогон)" << endl;  cout << "2 - генератор случайных чисел (рабочий прогон)" << endl;  }  //Функция считывает элементы массива, введённые пользователем  void readArray(unsigned int \*userValues, unsigned int \*valuesTest, int arraySize) {  for (int i = 0; i < arraySize; i++) {  cout << "Введите значение для элемента " << i << " : ";  cin >> userValues[i];  valuesTest[i] = userValues[i];  }  }  //Функция заполняет массив случайными значениями  void generateArray(unsigned int \*userValues, unsigned int \*valuesTest, int arraySize) {  for (int i = 0; i < arraySize; i++) {  userValues[i] = rand() % 65536;  valuesTest[i] = userValues[i];  }  }  int main(void) {  SetConsoleOutputCP(CP\_UTF8);  srand(time(NULL));  int menu = 1;  int workMode;  int arraySize;  unsigned int \*userValues;  unsigned int \*resultValues;  cout << "Практическая работа №1 'Поразрядные операции и их применение' Школьник Татьяна ИКБО-06-21" << endl;  while (menu != 0) {  cout << "Какое задание выполнить?" << endl;  cout << "1. Установить биты в единицу 0й, 3й, 11й" << endl;  cout << "2. Установить старшие 4 бита в ноль" << endl;  cout << "3. Умножить на 4" << endl;  cout << "4. Разделить на 4" << endl;  cout << "5. Обнулить разряд" << endl;  cin >> menu;  switch (menu) {  case 1:  onboading(&arraySize);  cin >> workMode;  userValues = new unsigned int[arraySize];  resultValues = new unsigned int[arraySize];  if (workMode == 1) {  readArray(userValues, resultValues, arraySize);  } else if (workMode == 2) {  generateArray(userValues, resultValues, arraySize);  } else { exit(0); }  cout << "\nУпражнение " << 1 << "\n" << endl;  for (int i = 0; i < arraySize; i++) {  cout << "Исходное значение " << endl;  printBinary(userValues[i]);  cout << endl << "Изменённое значение " << endl;  set1InBits(resultValues[i]);  printBinary(resultValues[i]);  cout << endl << endl;  resultValues[i] = userValues[i];  }  break;  case 2:  onboading(&arraySize);  cin >> workMode;  userValues = new unsigned int[arraySize];  resultValues = new unsigned int[arraySize];  if (workMode == 1) {  readArray(userValues, resultValues, arraySize);  } else if (workMode == 2) {  generateArray(userValues, resultValues, arraySize);  } else { exit(0); }  cout << "\nУпражнение " << 2 << "\n" << endl;  for (int i = 0; i < arraySize; i++) {  cout << "Исходное значение " << endl;  printBinary(userValues[i]);  cout << endl << "Изменённое значение " << endl;  set0ToBits(resultValues[i]);  printBinary(resultValues[i]);  cout << endl << endl;  resultValues[i] = userValues[i];  }  break;  case 3:  onboading(&arraySize);  cin >> workMode;  userValues = new unsigned int[arraySize];  resultValues = new unsigned int[arraySize];  if (workMode == 1) {  readArray(userValues, resultValues, arraySize);  } else if (workMode == 2) {  generateArray(userValues, resultValues, arraySize);  } else { exit(0); }  cout << "\nУпражнение " << 3 << "\n" << endl;  for (int i = 0; i < arraySize; i++) {  cout << "Исходное значение " << endl;  printBinary(userValues[i]);  cout << endl << "Изменённое значение " << endl;  multiply(resultValues[i]);  printBinary(resultValues[i]);  cout << endl << endl;  resultValues[i] = userValues[i];  }  break;  case 4:  onboading(&arraySize);  cin >> workMode;  userValues = new unsigned int[arraySize];  resultValues = new unsigned int[arraySize];  if (workMode == 1) {  readArray(userValues, resultValues, arraySize);  } else if (workMode == 2) {  generateArray(userValues, resultValues, arraySize);  } else { exit(0); }  cout << "\nУпражнение " << 4 << "\n" << endl;  for (int i = 0; i < arraySize; i++) {  cout << "Исходное значение " << endl;  printBinary(userValues[i]);  cout << endl << "Изменённое значение " << endl;  divide(resultValues[i]);  printBinary(resultValues[i]);  cout << endl << endl;  resultValues[i] = userValues[i];  }  break;  case 5:  onboading(&arraySize);  cin >> workMode;  userValues = new unsigned int[arraySize];  resultValues = new unsigned int[arraySize];  if (workMode == 1) {  readArray(userValues, resultValues, arraySize);  } else if (workMode == 2) {  generateArray(userValues, resultValues, arraySize);  } else { exit(0); }  cout << "\nУпражнение " << 5 << "\n" << endl;  for (int i = 0; i < arraySize; i++) {  if (workMode == 1) {  int n;  cout << "Введите номер бита (от 0 до 15): ";  cin >> n;  setZeros(resultValues[i], n);  } else {  int n = rand() % 16;  cout << endl << "Обнуляем бит под номером: " << n << endl;  setZeros(resultValues[i], n);  }  cout << "Исходное значение " << endl;  printBinary(userValues[i]);  cout << endl << "Изменённое значение " << endl;  printBinary(resultValues[i]);  cout << endl << endl;  resultValues[i] = userValues[i];  }  }  break;  }  system("pause");  return 0;  } |