|  |
| --- |
| **МИНОБРНАУКИ РОССИИ**  **САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  **ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  **«ЛЭТИ» ИМ. В. И. УЛЬЯНОВА (ЛЕНИНА)**  **Кафедра РЭС** |
| **ОТЧЕТ по лабораторной работе №1 по дисциплине «Информатика» Тема: Автоматизация перевода чисел из двоичной системы счисления в десятичную.** |
| |  |  | | --- | --- | | Студент гр. 1183 | Чаминов Д. А. | | Преподаватель | Ситников И. Ю. | |
| Санкт-Петербург  2021 |

**Содержание**

[Спецификация задания 3](#_Toc84611679)

[Описание алгоритма 4](#_Toc84611680)

[Блок-схема алгоритма 5](#_Toc84611681)

[Выбор и обоснование типов переменных 6](#_Toc84611682)

[Вводимые и выводимые параметры и их типы 7](#_Toc84611683)

[Структура проекта 8](#_Toc84611684)

[Текст программы 9](#_Toc84611685)

[Копии экрана при работе программы 10](#_Toc84611686)

[Сравнение с эталоном 11](#_Toc84611687)

[Результаты 13](#_Toc84611688)

# Спецификация задания

Разработанная программа должна отвечать следующим требованиям:

* Работать в операционной системе Windows
* Разрабатываться в среде MS Visual Studio на языке программирования C (Си)
* Программа должна по введённому пользователем десятичному числу (оно должно быть меньше, чем ) построить и вывести пользователю 32 двоичных разряда этого числа с ведущими нулями.

# Описание алгоритма

На вход программе с клавиатуры подается целое неотрицательное десятичное число, не превышающее .

Программа вычисляет каждый двоичный разряд: если число больше, чем , и меньше, чем , то двоичный разряд n этого числа равен 1, в противном случае – 0. Для корректности работы такой проверки необходимо начинать со старшего разряда и обнулять проверенные (это можно сделать при помощи взятия остатка от деления на или же вычитанием ). Таким образом, программа использует жадный алгоритм для представления исходного числа в виде суммы степеней 2:   
, где  равно двоичному разряду с номером (самый правый разряд имеет номер ).

При распечатке коэффициентов при от 31 до 0 получается двоичная запись числа с 32 разрядами, что и требуется от программы.

# Блок-схема алгоритма

Начало

p = p / 2

нет

Завершение

a = a - p

Вывод 0

нет

Ввод числа в a

p = 2^31

да

p > 0

да

a >= p

Вывод 1

# Выбор и обоснование типов переменных

Поскольку пользователь вводит целое неотрицательное число, которое представляется 32 двоичными разрядами, то для это числа достаточно 32 битного беззнакового целого

Поскольку максимальным значением p является , а минимальным – 0, для него подойдет тип unsigned int.

В программе присутствует вычитание беззнаковых, однако проверка, определяющее значение разряда исключает вероятность ошибки выхода значения за границы беззнаковых чисел. Умножения и сложения в программе нет, поэтому вероятность переполнения отсутствует.

# Вводимые и выводимые параметры и их типы

В программу с клавиатуры поступает единственное целое десятичное число, не превышающее .

Программа выводит строку – приглашение ко вводу (const char[22]), результат, содержащий строку – пояснение (const char[41]) и 32 символа const char, которые равны ‘0’ или ‘1’ – двоичное представление исходного числа. Затем программа выводит символ перевода каретки ’\n’ (const char).

# Структура проекта

Программа состоит из файла main.c, к которому подключаются заголовочные файлы <stdio.h>, отвечающий за функции стандартного ввода-вывода (scanf и printf), <stdlib.h>, отвечающий за функцию system, используемую в программе.

Структура каталога решения (вывод утилиты ‘tree’ с параметром ‘/F’):

───FromDecimalToBinary

main.c

FromDecimalToBinary.vcxproj

FromDecimalToBinary.vcxproj.filters

FromDecimalToBinary.vcxproj.user

# Текст программы

#define \_CRT\_SECURE\_NO\_WARNINGS // MS Visual Studio forbids using

// "unsafe" functions "scanf" and "printf"

#include <stdlib.h> // for 'system' requests

#include <stdio.h> // input-output library  
int main()

{

unsigned int a; // number that we should represent as binary

printf("Enter your number\n>>>"); // prompt for input

scanf("%d", &a); // getting 'a' value

unsigned int p = 0x80000000; // 2^n, n=31 the highest bit in number

printf("Your number in binary representation is "); // puts some beauty

while (p) { // while (p>=2^0), 2^0 - the lowest bit

if (a / p) { // checking current bit

printf("1"); // print this '1' bit

a %= p; // excludes this bit from a ( a &= !p ), p = 00...00100...0000

}

else {

printf("0"); // print this '0' bit

}

p /= 2; // decreasing p from 2^n to 2^(n-1)

}

putchar('\n'); // end of line (and output)

system("pause"); // do not close console, user should be able to see result

return 0; // EXIT\_SUCCESS

}

# Копии экрана при работе программы

Изображение выглядит как текст

Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

# Сравнение с эталоном

Для проверки надежности алгоритм был проверен на 1048576 случайных тестах (не исключено повторение некоторых из них).

Каждый тест был сгенерирован с использованием функции bin в языке программирования python3. Файл с тестами расположен в [моем хранилище google drive](https://drive.google.com/file/d/16S_gAy2GYiHgb0iiDewAmTU6o8xpPTLc/view?usp=sharing) (гиперссылка), под именем “TESTS”. В файле через пробел в каждой строке записана пара – входные и ожидаемые выходные данные для программы. Для удобства тестирования код программы был выведен в отдельную функцию, была написана программа, использующая эту функцию и проверяющая результат. Код программы приведен ниже.

#include <fstream>

#include <iostream>

void representAsBin(unsigned int a, char\* r)

{

unsigned int p = 0x80000000; // 2^n, n=31 the highest bit in number

char i = 0; // result index

while (p) { // while (p>=2^0), 2^0 - the lowest bit

if (a / p) { // checking current bit

r[i++] = '1';

a %= p; // excludes this bit from a ( a &= !p ), p = 00...00100...0000

}

else {

r[i++] = '0';

}

p /= 2; // decreasing p from 2^n to 2^(n-1)

}

}

int main()

{

std::ifstream fin("TESTS"); // open file stream, use "TESTS" file

while (fin.eof()) { // requests a method of ifstream, that

//returns true if we reach the end of file

char res[33], trueres[33]; // c-strings for result of algorithm and expected result

unsigned int a;

fin >> a >> trueres; // using 'operator>>' to get values for 'a' and 'trueres'

trueres[32] = '\0'; // end of c-string

res[32] = '\0'; // same

representAsBin(a, res); // run algorithm

if (res != trueres) { // check result

std::cout << "ERROR\nInput: " << a //using 'operator<<' for console output stream object

<< "\nExpected: " << trueres

<< "\nGot: " << res << '\n';

}

}

std::cout << "tests were passed successfully\n"; //same

return 0;

}

# Результаты

* Разработанное приложение работает на системах MS Windows.
* Приложение является консольным
* Исполняемый файл fromDecimalToBinary.exe имеет размер 9,00 КБ (9 216 байт)
* На ввод программе подается целое неотрицательное число меньшее, чем
* Проверка на случайных числах в допустимом диапазоне при помощи стандартной функции bin в языке python3 прошла успешно, ошибок не обнаружено
* Достоинства:
  + Быстрый перевод чисел (с тестами алгоритм справился быстрее, чем алгоритм, который их генерировал);
  + Понятный интерфейс программы.
* Недостатки:
  + Пользовательский интерфейс только на английском языке;
  + Ограничения на входные данные.