|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|

|  |  |  |
| --- | --- | --- |
|  |    |  |
| 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 ЛАБОРАТОРНОЙ РАБОТЕ № 5** |
| **по дисциплине** |
| **«**Процедурное программирование**»****Тема: «МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ. ПОДПРОГРАММЫ»** |
|  |
| Выполнил студент группы ИКБО-12-18  | Валяев Д.А. |
| Принял преподаватель | Миронов А.Н. |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «08» Февраля 2019 г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_201\_\_ г. |  *(подпись руководителя)* |

Москва 2018

Задание 1. Вычисление значения математического выражения

1. Условие задачи

Разработать функцию, которая вычисляет количество единиц в двоичном

коде целого числа.

2. Постановка задачи

Дано: одно целое число.

Ограничения на данные: число должно быть не больше 1023 и не меньше -1023.

Ограничения на решаемую задачу: нет

3. Математическая модель задачи

-Num – одно целое число

-формула для перевода в двоичную систему счисления:

while (num) {

 b += num % 2 \* k;

 num /= 2;

 k \*= 10;

 }

-далее в двоичном числе подсчитывается количество единиц и выводится результат

4. Определение входных и выходных данных

Входные данные одно целое число, вводится с клавиатуры.

Результат – количество единиц в двоичной записи введенного числа.

5. Таблица имен объектов программы

|  |  |  |  |
| --- | --- | --- | --- |
| *Имя объекта* | *Диапазон допустимых значений* | *Тип данных* | *Семантика* |
| num | -1023..1023 | int | Переменная, аргумент функции |
| u, k, y ,x ,b | -2 147 483 648 .. 2 147 483 647 | int | Переменная, аргумент функции |

6. Разработка алгоритма программы



7. Разработка тестов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Номер теста* | *Исходные данные* | *Эталон результата* | *Результат программы* | *Отметка о правильном прохождении теста* |
| num | … | u | u |
| 1. | 1000 | … | 6 | 6 | Пройден |
| 2. | 2 | … | 1 | 1 | Пройден |
| 3. | 555 | … | 5 | 5 | Пройден |
| 4. | 15 | … | 4 | 4 | Пройден |
| 5. | 98 | … | 3 | 3 | Пройден |

8. Исходный код программы

#include "pch.h"

#include <locale.h>

#include <iostream>

#include <string>

#include <cstdlib>

#include <sstream>

using namespace std;

auto f(int num) {

 int b = 0, k = 1;

 while (num) {

 b += num % 2 \* k;

 num /= 2;

 k \*= 10;

 }

 auto str = std::to\_string(b);

 int y = str.length();

 int x = 0;

 int u = 0;

 while (x < y) {

 if (str[x] == '1')

 u += 1;

 x++;

 }

 return u;

}

int main() {

 setlocale(LC\_CTYPE, "RUS");

 cout << "Введите числo: ";

 int num;;

 cin >> num;

 f(num);

 cout << "Кол-во единиц в двоичном представлении числа: " <<f(num) << endl;

}

Задание 2. Разработка программы с применением математической модели

1. Условие задачи

Разработать процедуру, которая в последовательности целых чисел

находит число, в двоичном представлении которого больше всего единиц и

количество единиц в этом числе.

2. Постановка задачи

Дано: множество целых чисел.

Ограничения на данные: числа должны быть не больше 1023 и не меньше -1023.

Ограничения на решаемую задачу: нет.

3. Математическая модель задачи

-str – множество целых чисел

-формула для выделения чисел из строки

stringstream ss(str);

 int num, one, max = -1, Mnum;

 while (ss >> num) {

 one = f(num);

 if (one > max) {

 max = one;

 Mnum = num;

 }

 }

-формула для перевода в двоичную систему счисления:

while (num) {

 b += num % 2 \* k;

 num /= 2;

 k \*= 10;

 }

-далее в двоичном числе подсчитывается количество единиц и выводится результат

4. Определение входных и выходных данных

Входные данные – множество целых чисел, вводится с клавиатуры.

Результат – число в двоичном представлении которого наибольшее количество единиц и количество единиц в этом числе.

5. Таблица имен объектов программы

|  |  |  |  |
| --- | --- | --- | --- |
| *Имя объекта* | *Диапазон допустимых значений* | *Тип данных* | *Семантика* |
| str | 4294967294 | string | Переменная, аргумент функции |
| u, k, y ,x ,b, one, max, Mnum | -2 147 483 648 .. 2 147 483 647 | int | Переменная, аргумент функции |
| num | -1023..1023 | int | Переменная, аргумент функции |

6. Разработка алгоритма программы



7. Разработка тестов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Номер теста* | *Исходные данные* | *Эталон результата* | *Результат программы* | *Отметка о правильном прохождении теста* |
| str | … | z | z |
| 1. | 2 3 4 5 | … | 4..15 | 4..15 | Пройден |
| 2. | 12 131 4 | … | 3..131 | 3..131 | Пройден |
| 3. | 2 3  | … | 2..3 | 2..3 | Пройден |
| 4. | 77 4 12 9 | … | 4..77 | 4..77 | Пройден |
| 5. | 100 5 0 | … | 3..100 | 3..100 | Пройден |

8. Исходный код программы

#include "pch.h"

#include <locale.h>

#include <iostream>

#include <string>

#include <cstdlib>

#include <sstream>

using namespace std;

auto f(int num) {

 int b = 0, k = 1;

 while (num) {

 b += num % 2 \* k;

 num /= 2;

 k \*= 10;

 }

 auto str = std::to\_string(b);

 int y = str.length();

 int x = 0;

 int u = 0;

 while (x < y) {

 if (str[x] == '1')

 u += 1;

 x++;

 }

 return u;

}

void h(const string &str) {

 stringstream ss(str);

 int num, one, max = -1, Mnum;

 while (ss >> num) {

 one = f(num);

 if (one > max) {

 max = one;

 Mnum = num;

 }

 }

 cout << "Максимальное количество единиц: " << max << " в числе: " << Mnum << endl;

}

int main() {

 setlocale(LC\_CTYPE, "RUS");

 cout << "Введите числа(разделитель-пробел): ";

 string str;

 getline(cin, str);

 h(str);

}

Задание 3. Поразрядные операции

1. Условие задачи

Дана последовательность из n целых чисел. Определить количество чисел,

содержащих в своей десятичной записи хотя бы одну цифру 7 и являющихся

простыми числами.

2. Постановка задачи

Дано: последовательность целых чисел.

Ограничения на данные: числа должны быть не больше 2 147 483 647 и не меньше -2 147 483 648.

Ограничения на решаемую задачу: нет.

3. Математическая модель задачи

-str – множество целых чисел

-формула для выделения чисел из строки

stringstream ss(str);

 int num, one, max = -1, Mnum;

 while (ss >> num) {

 one = f(num);

 if (one > max) {

 max = one;

 Mnum = num;

 }

 }

- далее подсчитывается количество чисел, имеющих в своей записи цифру 7.

4. Определение входных и выходных данных

Входные данные – множество чисел, вводятся с клавиатуры.

Результат – количество чисел, имеющих в своей записи цифру 7.

5. Таблица имен объектов программы

|  |  |  |  |
| --- | --- | --- | --- |
| *Имя объекта* | *Диапазон допустимых значений* | *Тип данных* | *Семантика* |
| u, k, y ,x ,b, one, max, Mnum, num | -2 147 483 648 .. 2 147 483 647 | int | Переменная, аргумент функции |
| str | 4294967294 | string | Переменная, аргумент функции |

6. Разработка алгоритма программы



7. Разработка тестов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Номер теста* | *Исходные данные* | *Эталон результата* | *Результат программы* | *Отметка о правильном прохождении теста* |
| str | … | z | z |
| 1. | 7 7 7 7 | … | 4 | 4 | Пройден |
| 2. | 13 777 8 4 | … | 1 | 1 | Пройден |
| 3. | 4 8 34 17 | … | 1 | 1 | Пройден |
| 4. | 45 78 37 7 | … | 3 | 3 | Пройден |
| 5. | 8 4 33 6 4 | … | 0 | 0 | Пройден |

8. Исходный код программы

#include "pch.h"

#include <locale.h>

#include <iostream>

#include <string>

#include <cstdlib>

#include <sstream>

using namespace std;

auto f(int num) {

 int b = num;

 auto str = std::to\_string(b);

 int y = str.length();

 int x = 0;

 int u = 0;

 while (x < y) {

 if (str[x] == '7')

 u++;

 if (u == 1)

 break;

 x++;

 }

 return u;

}

auto h(const string &str) {

 stringstream ss(str);

 int num, one, er = 0;

 while (ss >> num) {

 one = f(num);

 er = er + one;

 }

 return er;

}

int main() {

 setlocale(LC\_CTYPE, "RUS");

 cout << "Введите числа(разделитель-пробел): ";

 string str;

 getline(cin, str);

 h(str);

 cout << "количество 7: " << h(str) << endl;

}