

Домашнее задание по «Программной инженерии» №1.

Студентка: Ковалёва Дарья, СГНЗ-34Б, 8 вариант.

Постановка задачи:

В текстовом файле содержатся целые числа. Найти сумму цифр каждого числа и если она четная, то перевести его в двоичную систему счисления, а если нечетная, то преобразовать число в обратном порядке следования цифр. Результат вывести на экран.

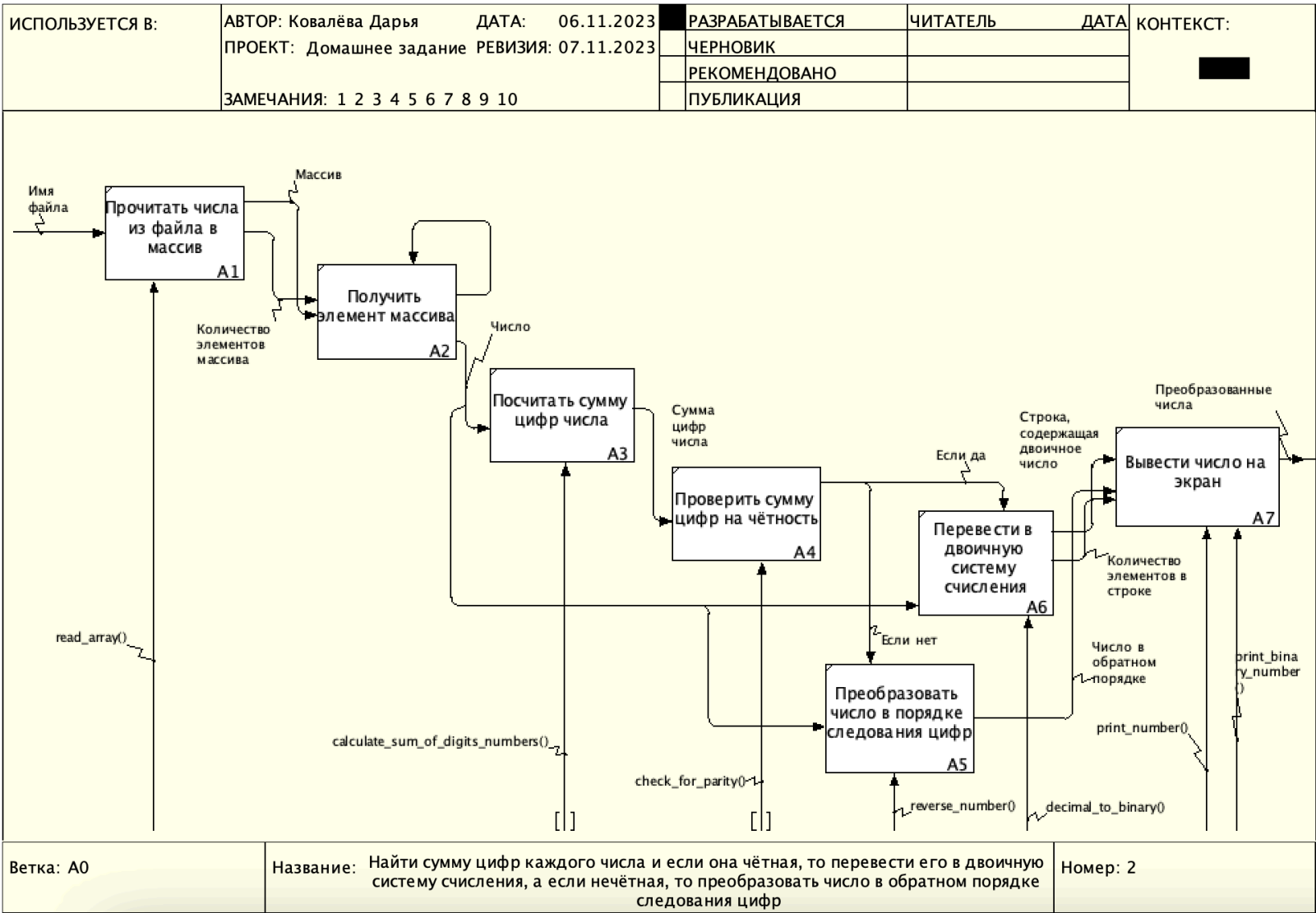
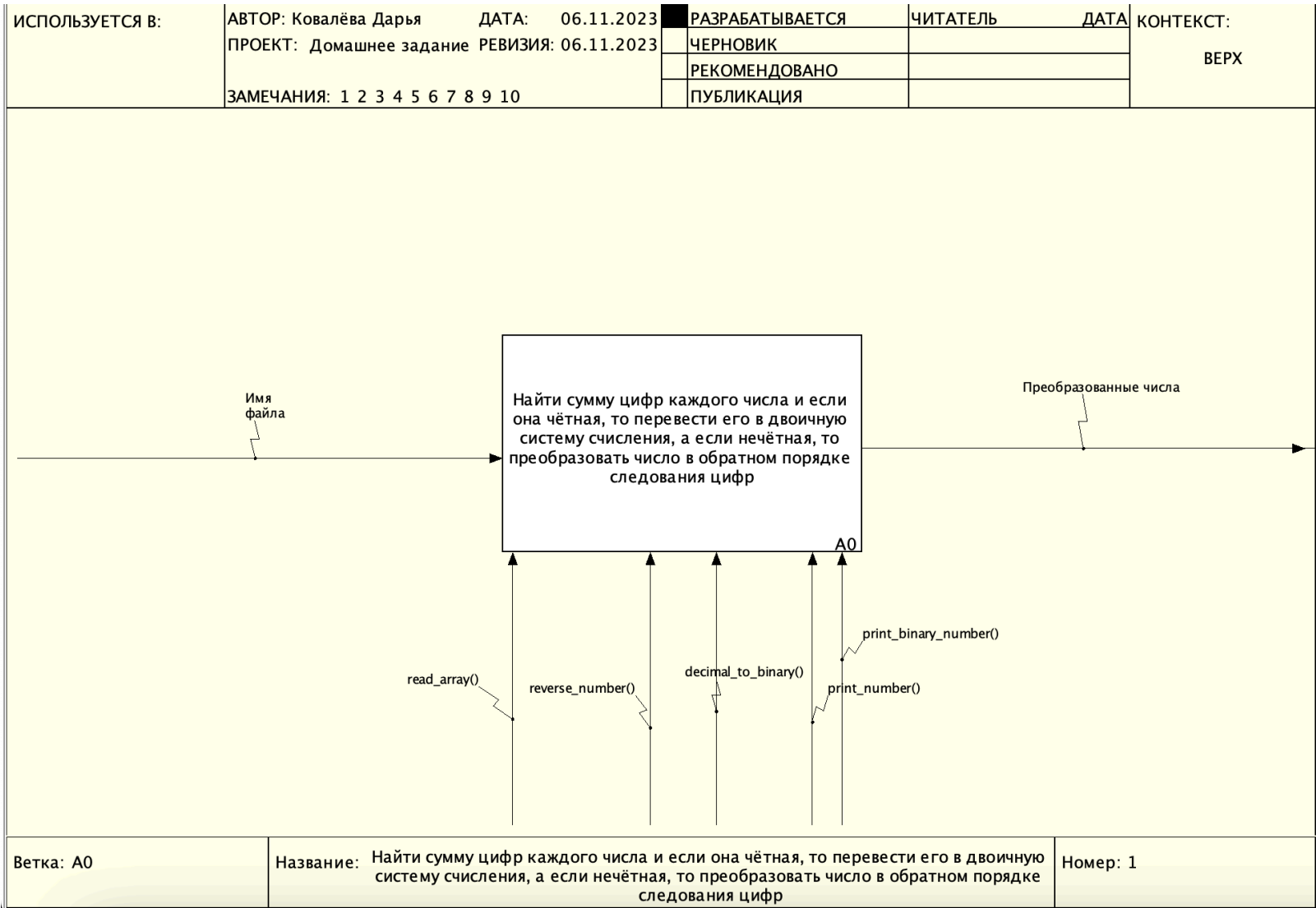
Допущения:

- 1) В файле только целые числа, которые записаны правильно
- 2) Числа разделяются произвольным количеством пробелов
- 3) Считывается все числа из файла в массив, количество чисел в файле не превышает 568
- 4) Отрицательные числа преобразуются по модулю: **при чётной сумме цифр** - первый бит выделяется под знак, если число положительное, то первый бит = 0, если число отрицательное, то первый бит = 1, **при нечётной сумме цифр** - преобразовать число в обратном порядке следования цифр по модулю
- 5) Каждое число ограничено целым типом

Примеры работы программы

Исходный файл	Вывод программы
Несуществующий файл	Ошибка! Файла не существует!
Пустой файл	Ошибка! В файле нет данных!
Некорректный файл: . 3 5иии	Ошибка! В файле нет корректных данных!
10 11 12 -13 14 -15 -16	1 01011 21 11101 41 11111 -61
10 -12 14 -16 18 21	1 -21 41 -61 81 12
11 -13 15 -17 19 20	01011 11101 01111 110001 010011 010100

Контекстная диаграмма:



Спецификация модуля A1:

Имя	read_array()
Функция	Считать числа из файла в массив
Список параметров	Файловая переменная, массив, количество элементов в массиве
Входные данные	Файловая переменная
Выходные данные	Массив, количество элементов в массиве
Внешние эффекты	Переменная указателя файла на следующую строку

Тестовые данные для модуля A1:

Строка файла	Результат
Пустая	Пустой массив, количество элементов = 0
10	10, количество элементов = 1
1 10 12	1, 10, 12, количество элементов = 3

Примечание: для тестирования модуля выделяется два класса эквивалентности: пустая строка и не пустая. При этом для не пустой строки рассматриваются случаи, когда элемент один и когда их несколько.

Псевдокод для модуля A1:

read_array

Вход: file – файловая переменная

Выход: array – массив целых чисел, n – количество элементов в массиве

n = 0

Пока не конец файла **делать**

 считать элемент в array[n]

 увеличить n на 1

Все пока

Спецификация модуля А3:

Имя	calculate_sum_of_digits_numbers()
Функция	Считает сумму цифр числа
Список параметров	Число, сумма цифр числа
Входные данные	Число
Выходные данные	Сумма цифр числа
Внешние эффекты	Нет

Тестовые данные для модуля А3:

Число	Результат
10	1
-13	4

Примечание: для тестирование данного модуля выделяется два класса эквивалентности: положительное число и отрицательное число. Для отрицательных чисел берётся сумма цифр по модулю.

Псевдокод для модуля А3:

calculate_sum_of_digits_numbers()

Вход: num – число

Выход: sum – сумма цифр числа

num = abs(num)

sum = 0

Пока num не равно 0 **делать**

 sum += num % 10;

 num /= 10;

Все пока

Спецификация модуля A4:

Имя	check_for_parity()
Функция	Проверяет число на чётность
Список параметров	Число, признак
Входные данные	Число
Выходные данные	Признак
Внешние эффекты	Нет

Тестовые данные для модуля A4:

Число	Результат (1 - True, 0 - False)
Нет числа	
10	False
15	True

Примечание: для тестирование данного модуля выделяется два класса эквивалентности: нет числа и число есть. При этом там где число есть рассматривается два случая, когда сумма цифр числа нечётная и сумма цифр числа чётная.

Псевдокод для модуля A4:

check_for_parity()

Вход: num – число

Выход: is_even – признак

Пока есть sum **делать**

 Если sum % 2 == 0

 is_even = 1

 Иначе

 is_even = 0

Все пока

Спецификация модуля A5:

Имя	reverse_number()
Функция	Преобразует число в обратном порядке следования цифр
Список параметров	Число, число в обратном порядке
Входные данные	Число
Выходные данные	Число в обратном порядке
Внешние эффекты	Нет

Тестовые данные для модуля A5:

Число	Результат
-12	-21
140	41

Примечание: для тестирование данного модуля выделяется два класса эквивалентности: положительное число и отрицательное число. При этом для отрицательного числа программа преобразует цифры в обратном порядке следования цифр по модулю и знак минус ставит впереди записи преобразованного числа.

Псевдокод для модуля A5:

reverse_number()

Вход: num – число

Выход: reversed_num – число в обратном порядке

reversed_num = 0,

Пока num не равно 0 делать

```
    reversed_num = reversed_num * 10 + num % 10;  
    num /= 10;
```

Все пока

Спецификация модуля А6:

Имя	decimal_to_binary()
Функция	Переводит число в двоичную систему счисления
Список параметров	Число, строка, содержащая двоичное число, количество элементов строки
Входные данные	Число
Выходные данные	Строка, содержащая двоичное число, количество элементов строки
Внешние эффекты	Нет

Тестовые данные для модуля А6:

Число	Результат
-11	11011
11	1011

Примечание: для тестирование данного модуля выделяется два класса эквивалентности: положительное и отрицательное число. При этом для положительного числа первый бит будет = 0, а для отрицательного первый бит = 1.

Псевдокод для модуля А6:

decimal_to_binary()

Вход: num – число

Выход: result – строка, содержащая двоичное число, size – количество элементов строки

binary[32], i = 0

Пока num не равно 0 делать

Если num > 0

sign = '0'

Иначе

sign = '1'

Пока num > 0 делать

binary[i] = num % 2

num /= 2

увеличить i на 1

Все пока

```
result[0] = sign
```

```
j = i - 1
```

Пока $j \geq 0$ делать

 записать двоичную цифру `binary[j]` в `result`

 уменьшить `j` на 1

Все пока

Спецификация модуля A7:

Имя	print_number()
Функция	Выводит число в консоль
Список параметров	Число в обратном порядке, преобразованные числа
Входные данные	Число в обратном порядке
Выходные данные	Нет
Внешние эффекты	Нет

Тестовые данные для модуля A7:

Число	Результат
-10	-10
10	10

Примечание: для тестирования модуля выделяется один класс эквивалентности и рассматриваются два случая, когда число отрицательное и положительное.

Псевдокод для модуля A7:

print_number()

Вход: num – число

Выход: –

Пока есть элемент num **делать**

 вывести num

Все пока

Спецификация модуля A7:

Имя	print_binary_number()
Функция	Выводит строку в консоль
Список параметров	Строка, содержащая двоичное число, количество элементов строки
Входные данные	Строка, содержащая двоичное число, количество элементов строки
Выходные данные	Нет
Внешние эффекты	Нет

Тестовые данные для модуля A7:

Строка	Результат
10111	11101
101101	101101

Примечание: для тестирования модуля выделяется один класс эквивалентности и рассматривается строка с двоичным представлением числа.

Псевдокод для модуля A7:

print_binary_number()

Вход: binary[] – строка, size – количество элементов строки

Выход: –

j = 0

Пока j < size **делать**

 печатать binary[j]

 увеличить j на 1

Все пока

Псевдокод основной программы:

file – файловая переменная

size – количество успешно считанных чисел из файла

numbers – массив целых чисел

is_even – флаг, который указывает на чётность или нечётность суммы цифр числа

numbers[I] – элемент массива

binary – строка, содержащая двоичное представление числа

reversed_number – переменная, содержащая обратное представление числа

открыть file на чтение

в size = read_array(file, numbers);

закрыть file

i = 0

Пока i < size **делать**

 is_even = check_for_parity(numbers[i]);

Если is_even == 1

 в binary = decimal_to_binary(numbers[i]);

 напечатать binary

Иначе

 в reversed_number = reverse_number(&numbers[i])

 print_number(&reversed_number)

 увеличить i на 1

Все пока