Щербатюк Дарья, ИУ7-34

Типы и структуры данных

Лабораторная работа 1

Обработка больших чисел

Вариант 2

**1.Техническое задание**

1.1 Задача: Смоделировать операцию умножения действительного числа в формате +-m.n E +-K (суммарная длина мантиссы (m+n) - до 30 значащих цифр, а величина порядка К - до 5 цифр) на целое число длиной до 30 десятичных цифр. Результат выдать в форме +- 0.m1 E +-K1, где m1 - до 30 значащих цифр, К1 - до 5 цифр.

1.2 Входные данные: строка, содержащая вещественное число в виде [-]m.n e[-]k. Суммарная длина мантиссы m+n – до 30 цифр, порядка k – до 5 цифр. Допускается отсутствие точек, e, знаков. После е необходимо ввести порядок.

Строка, содержащая целое число в виде [-]t, длина t – до 30 цифр.

1.3 Выходные данные: «большое число», нормализованное в виде [-]0.m1 e[-]k1, длина m1 – до 30 цифр, k1 – до 5.

Функция программы: умножение вещественого числа на целое;

1.4 Аварийные ситуации

1. Некорректный ввод строки с числом

Программно заблокирована возможность вводить что-либо кроме символов цифр, “-”, “+”, “e”, “.”. Если строка так или иначе содержит некорректную запись (например 12-1), будет выведено сообщение об ошибке.

1. Превышение допустимого количества цифр числа в мантиссе.

Будет выведено сообщение об ошибке.

1. Порядок числа при вводе >99999 или <-99999

Будет выведено сообщение об ошибке.

1. Порядок вводимого числа в допустимых пределах, но в результате операции происходит его переполнение.

Будет произведена попытка деления; вместо результата будет выведена строка <Порядок вышел за границы>.

**2.Алгоритм**

0. Считывание и обработка целого числа

1. Запись знака в int
2. Запись самого числа в array of integer
3. Запись порядка в integer
4. Считывание и обработка вещественного числа
5. Запись знака в int
6. Запись мантиссы в array of integer
7. Запись порядка в integer
8. Выполнение умножения
9. Запись знака результата в int
10. Обработка мантиссы результата
11. Печать результата

**3.Вутренние структуры данных**

**struct chislo**

{

int znak\_mantis;// -1 - отрицательное 1 - положительное

int mantis[LEN\_MANTIS];

int poriadok;

};

**7.Функции**

***Функции чтения и обработки числа***

*int read\_chislo(struct chislo \*chislo)*

Параметры:

*struct chislo \*chislo – структура хранящая число.*

***Псевдокод функции:***

*Пока идет ввод символов.*

*Если символ является верным*

*Записать его в строку*

*Если длинна строки превысила допустимое значение*

*Выдать ошибку некорректного ввода*

Если ничего не введено

Выдать ошибку пустого ввода

Если ввод прошел удачно

Проверить количество вхождений символов '.' '-' '+' 'E'

Если количество вхождений пройдено

Взять мантиссу (функция int take\_mantis(char \*str,int \*mantis,int \*poriadok,int \*znak\_mantis) )

Если взятие мантиссы прошло верно

Взять порядок (функция int take\_poriadok(char \*str,int \*poriadok) )

Если взятие порядка прошло верно

Заполнить незначащие нули -1 для удобного вывода числа

Вернуть код ошибки

***Функция умножения двух чисел***

int multiplicate(struct chislo \*chislo1,struct chislo chislo2)

Параметры:

*struct chislo \*chislo1 – структура хранящая первое число. В нее в конце запишем ответ*

*struct chislo \*chislo2 – структура хранящая второе число.*

Псевдокод функции:

*Перемножаем знаки мантисс двух чисел.*

*Полученный результат записываем в первое число.*

*Складываем порядки.*

*Полученный результат записываем в первое число.*

*Проверяем порядок на переполнение*

*Если переполнения нет*

*Начинаем перемножение по аналогии с умножением в столбик*

*Перемножаем последовательно каждое число второго числа на первое*

*И прибавляем в промежуточный массив учитывая порядки*

*Массив мантиссы первого числа приравниваем к промежуточному*

*Проверяем нулевой результат или нет*

*Проверяем на переполнение*

*Возвращаем ошибку*

**6.Тесты**

|  |  |  |
| --- | --- | --- |
| **Вещественное** | **Целое** | **Результат** |
| **[31 цифра]** |  | **ERROR: overflow mantis.** |
| **123.4** | **50 цифр** | **ERROR: overflow mantis.** |
| **123.4** | **0** | **+0. E +0** |
| **123.45 E -99999** | **-123** | **-0.1518435 E -99994** |
| **1234.56 E 99999** | **1234** | **ERROR: owerflow.** |
| **123.4** | **1** | **+0.1234 E +3** |
| **1** | **3** | **+0.3 E +1** |
| **123w123** |  | **ERROR: uncorrect input.** |
| **123E+-1** |  | **ERROR: uncorrect input.** |
| **123w** |  | **ERROR: uncorrect input.** |
| **+8** | **+4** | **+0.32 E +2** |
| **-8** | **-4** | **+0.32 E +2** |
| **-40** | **20** | **-0.8 E +3** |
| **50** | **-25** | **-0.125 E +4** |
| **123.12.2** |  | **ERROR: uncorrect input.** |
| **999..9[30 цифр]** | **2** | **+0.199999999999999999999999999998 E +30** |
|  |  |  |
|  |  |  |

**7.Ответы на вопросы**

1. Каков возможный диапазон чисел, представляемых в ПК?

Для 64-разрядного процессора максимально возможным значением целого числа является 2^64-1 = 18446744073709551615, а минимально возможным -2^64 = -18446744073709551616. Для действительного числа, на которое отводится 52 разряда под мантиссу и 11 под порядок, граничными значениями являются 3.6E–4951 и 1.1E+4932.

2. Какова возможная точность представления чисел, чем она определяется?

В первую очередь, точность определяется способом представления действительного числа. Представление в нормализованном виде, когда мантисса лежит в диапазоне [0.1, 1) обеспечивает максимально возможное количество хранимых чисел мантиссы, а следовательно, максимальную точность. При том, что под мантиссу отводится 52 разряда, количество цифр в ней не может превышать 16-ти.

3. Какие стандартные операции возможны над числами?

Над числами возможны такие операции : сложение, вычитание, умножение, деление, а также операция сравнения.

4. Какой тип данных может выбрать программист, если обрабатываемые числа превышают возможный диапазон представления чисел в ПК?

Для обработки больших чисел программист может выбрать тип данных, называемый structure в языке C.

5. Как можно осуществить операции над числами, выходящими за рамки машинного представления?

Обеспечить их представление, используя поля структуры (записи), и реализовать операцию по тому же алгоритму, как она делается вручную на бумаге.