

БЛОК-3 (СТЕК)

(обязательную задачу сдать до 9 апреля)

Задача 1 “Алгебраическая сумма” (обязательная).

Для ввода задана последовательность символов, представляющая собой (без ошибок) формулу следующего вида:

<формула> ::= <цифра> | (<формула> <знак> <формула>)
<знак> ::= + | -
<цифра> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Вычислить значение этой формулы. Пример: ((5-2)+7). → 10)
Считать, что запись формулы завершается точкой.

Идея.

Считываем в цикле символ за символом (пока не точка). Открывающие скобки пропускаем (игнорируем). Цифры и знаки ‘+’ или ‘-’ заносим в стек. Встретив закрывающую скобку, сразу же извлекаем из стека тройку <цифра><‘+’> или <‘-’><цифра> (путем чтения из стека трёх верхних элементов – двойных слов), вычисляем соответствующую ей формулу, а результат помещаем в стек. Возобновляем процесс чтения символов (конец – прочитана точка – тогда ответ извлекаем из стека и выводим его на экран). Не забываем преобразовывать вовремя код прочитанной цифры в её числовое значение.

Учесть, что при нормальной работе со стеком команды **push** и **pop** работают только с двойными словами => данные записываем в стек и считываем из стека в 32-битовом формате. Поэтому для ввода символа используем байтовый регистр, например, **inchar BL**, а для работы со стеком используем полный регистр, например, **push EBX** или **pop EBX**.

Задача 2 “Max/Min” (дополнительная 10 очков).

Для ввода задана последовательность символов, представляющая собой (без ошибок) формулу следующего вида:

<формула> ::= <цифра> | M(<формула>, <формула>) | m(<формула>, <формула>)
<цифра> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

где **M** (латинская!) трактуется как max (максимум), а **m** – как min (минимум).

Вычислить значение этой формулы. Пример: M(2, m(5, 7)). → 5
Считать, что запись формулы завершается точкой.

Идея.

Считываем в цикле символ за символом (пока не точка). Запятые и открывающие скобки пропускаем (игнорируем). Буквы ‘M’, ‘m’ или цифру заносим в стек. Встретив закрывающую скобку, сразу же извлекаем из стека тройку <‘M’> или <‘m’><цифра><цифра> (путем чтения из стека трёх верхних элементов – двойных слов), вычисляем соответствующую ей формулу, а результат помещаем в стек. Возобновляем процесс чтения символов (конец – прочитана точка – тогда извлекаем ответ из стека и выводим его на экран)

Задача 3. “Округление” (дополнительная – 30 очков)

Дана запись неотрицательного вещественного числа X произвольной длины и беззнаковое число m , укладываемое в диапазон байта (от 0 до 255).

Вещественное число X имеет следующий вид:

$\langle \text{вещественное число} \rangle ::= \langle \text{последовательность} \rangle . \langle \text{последовательность} \rangle$

$\langle \text{последовательность} \rangle ::= \langle \text{цифра} \rangle | \langle \text{цифра} \rangle \langle \text{последовательность} \rangle$

$\langle \text{цифра} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

Требуется вывести число X с m цифрами после точки, соблюдая следующие условия:

- 1) при отбрасывании лишних цифр в дробной части следует выполнить округление числа X с учётом значения $(m+1)$ -ой цифры после точки (значения всех остальных цифр справа от $(m+1)$ -ой цифры - игнорируются);
- 2) если в дробной части числа X недостаточно цифр (меньше m) - дополнить недостающие цифры нулями при выводе;
- 3) если число записано неверно – прекратить работу программы с диагностикой: *ошибка в записи числа*;
- 4) Ввод числа X осуществлять посимвольно (по макрокоманде **inchar**), конец записи числа – пробел; вывод преобразованного числа – посимвольно (по макрокоманде **outchar**);
- 5) Ввод значения m – числовой (по макрокоманде **inint**).
- 6) Массивы в решении использовать запрещено!

Примеры:

$X = 0.1234567890, m = 3 \rightarrow 0.123$

$X = 0.1234567890, m = 4 \rightarrow 0.1235$

$X = 0.01234567890, m = 4 \rightarrow 0.0123$

$X = 0.4999999999, m = 2 \rightarrow 0.50$

$X = 0.4999999999, m = 5 \rightarrow 0.50000$

$X = 0.4123999999, m = 5 \rightarrow 0.41240$

$X = 4999.4123999999, m = 0 \rightarrow 4999$

$X = 4999.99999999, m = 0 \rightarrow 5000$

$X = 1.23, m = 10 \rightarrow 1.2300000000$

$X = 0.0, m = 10 \rightarrow 0.0000000000$

$X = 99999.999999, m=4 \rightarrow 100000.0000$

Ваша программа должна обрабатывать число точно так же, как его обрабатывает во Free Pascal процедура форматного вывода вещественного числа:

Writeln(X: 1: m)

Рекомендации. Заметим, что число X *может быть полностью изменено* вследствие округления, а его обработка «на лету» невозможна в силу того, что число m приходит на вход после числа X . Таким образом, запись числа X необходимо сохранить в памяти. Учитывая, что длина записи числа может быть произвольной, для сохранения следует использовать **стек** (так как мы не можем выделить в массиве заранее столько памяти, чтобы сохранить всю запись числа). Следовательно, обработку числа следует тоже проводить в стеке, правильно настроив регистры-модификаторы на нужные места числа (начало его целой и дробной частей). Обратите внимание, что многие цифры после точки могут не повлиять на ответ, но, тем не менее, они должны быть сохранены в стеке.

Задача 4. “ Рассвет ” (дополнительная – максимально 50 очков)

На вход поступает непустая последовательность натуральных чисел, представляющих **высоты домов** (под **высотой дома** понимается количество этажей в нём). Конец последовательности - число **0**.

Дома перечисляются *в направлении с запада на восток* (то есть *слева направо*) и выстроены так, что если *восточнее* (то есть *правее*) от *данного дома* есть дом более высокий или равный ему по высоте, то из окон *данного дома* не виден рассвет. Нумерация домов начинается от единицы *с запада на восток*.

Вывести на экран последовательность домов (в формате *высота_дома : номер_дома*), **из которых виден рассвет**, в следующем порядке:

- а)** от самого *восточного* дома до самого *западного* (то есть *справа налево*) **(20 очков)**.
- б)** от самого *западного* дома до самого *восточного* (то есть *слева направо*) **(30 очков)**.

Примечание:

Номера домов лежат в диапазоне от **1** до **255**

Высоты домов лежат в диапазоне от **1** до **255**

Примеры (# - обозначение этажа в доме):

-----Пример 1-----

```
Ввод:      6 5 4 5 3 2 4 1
;          #
;          # #   #
;          # # # #   #
;          # # # # #   #
;          # # # # # # #
;          # # # # # # #
;номера: 1 2 3 4 5 6 7 8
```

Вывод для а):

```
1 8
4 7
5 4
6 1
```

Вывод для б):

```
6 1
5 4
4 7
1 8
```

-----Пример 2-----

```
Ввод:      5
;          #
;          #
;          #
;          #
;          #
;          #
;номера: 1
```

Вывод для а):

```
5 1
```

Вывод для б):

```
5 1
```

-----Пример 3-----

```
Ввод:      4 3 1
;          #
;          # #
;          # #
;          # # #
;номера: 1 2 3
```

```
Вывод для а):
1 3
3 2
4 1
```

```
Вывод для б):
4 1
3 2
1 3
```

-----Пример 4-----

```
Ввод:      4 4 4 3
;          # # #
;          # # # #
;          # # # #
;          # # # #
;номера: 1 2 3 4
```

```
Вывод для а):
3 4
4 3
```

```
Вывод для б):
4 3
3 4
```

-----Пример 5-----

```
Ввод:      2 2 2
;          # # #
;          # # #
;номера: 1 2 3
```

```
Вывод для а):
2 3
```

```
Вывод для б):
2 3
```

-----Пример 6-----

```
Ввод:      3 2 1 6
;          #
;          #
;          #
;          #
;          # #
;          # # #
;номера: 1 2 3 4
```

```
Вывод для а):
6 4
```

```
Вывод для б):
6 4
```

Требования к решению:

- 1) запрещено описывать массивы;
- 2) следует использовать стек для хранения высот домов и их номеров;
- 3) после вывода ответа стек привести в то состояние, в каком он был до начала чтения чисел (для проверки этого: выводить значение **ESP** до начала чтения чисел и после вывода ответа)
- 4) в программе должно быть достаточное количество *комментариев*, поясняющих алгоритм решения задачи.