

Логическая упорядоченная запись данных LEDATA

Эта запись содержит связанные данные, из которых сборщик затем строит часть образа памяти микропроцессора 8086.

Общий формат записи следующий:

///					
rec	record	segment	enumerated	dat	chk
typ	length	index	data		sum
A0h			offset		
___	_____	___///___	_____	___	___
				rpt	

- segment index
Это поле (индекс сегмента) не может быть нулевым и содержит индекс, ссылающийся на одну из записей SEGDEF, предшествующих этой записи LEDATA (Индексация сегментов начинается с единицы).
- enumerated data offset
Это поле (смещение упорядоченных данных) содержит смещение упорядоченных данных относительно ЛСЕТ, указанного индексом сегмента, т.е. смещение первого байта данных из поля dat.
- dat
Это поле содержит до 1024 последовательных байт перемещаемых или абсолютных данных, в противном случае необходимо записать данные в несколько записей LEDATA.
- chksum
Последним полем в каждой записи является поле chksum, содержащее дополнение до двух суммы (по модулю 256) всех остальных байт в записи.

Рассмотрим на примере программы 3.asm:

```
dseg segment
N dw 34
hi db 'Hello, world!!!'
a dw 10h,-3
dseg ends

sseg segment stack
db 'bye'
sseg ends

end
```

Программа OBJVIEW для файла 3.OBJ выдает следующий листинг:

```
A0 19 00 01 00 00 22 00 48 65 6C 6C 6F 2C 20 77 a.....".Hello, w
6F 72 6C 64 21 21 21 10 00 FD FF 4D orld!!!..M
A0:
Логическая упорядоченная запись данных (LEDATA)
19 00:
Длина записи 25 байт(a).
01:
Индекс сегмента: 01
00 00:
Смещение от начала сегмента 0 байт(a).
4D:
Контрольная сумма (Chksum): 77.
```

```
A0 07 00 02 00 00 62 79 65 17          a.....bye.
A0:
  Логическая упорядоченная запись данных (LEDATA)
07 00:
  Длина записи 7 байт (a).
02:
  Индекс сегмента: 02
00 00:
  Смещение от начала сегмента 0 байт (a).
17:
  Контрольная сумма (Chksum): 23.
```

Требуется написать программу, которая обрабатывает программу на языке ассемблера и формирует логические упорядоченные записи данных LEDATA в виде шестнадцатиричного дампа.

Исходные данные

Данные вводятся со стандартного устройства ввода. Пустые строки в программе отсутствуют. Программа состоит только из сегментов данных, каждый из которых содержит директивы db и/или dw. Все сегменты имеют различные имена. Длина сегментов - не более 65536 байтов. Разделителями лексем могут служить пробелы и символы табуляции. Пробелы в конце строк отсутствуют.

Выходные данные

Данные выводятся на стандартное устройство вывода - по одной строке на каждый сегмент. Пробелы в конце строки не допускаются.

Пример исходных данных:

```
dseg segment
N      dw 34
hi     db 'Hello, world!!!'
a      dw 10h,-3
dseg  ends
sseg  segment stack
      db 'bye'
sseg  ends
end
```

Пример выходных данных:

```
A0 19 00 01 00 00 22 00 48 65 6C 6C 6F 2C 20 77 6F 72 6C 64 21 21 21 10 00 FD FF 4D
A0 07 00 02 00 00 62 79 65 17
```