

## БЛОК-4 (часть 1)

### «Процедуры нерекурсивные: передача параметров через стек»

#### Задача 1. Частоты вхождений букв (обязательная)

Описать в программе следующие данные:

```
.data  
N equ 100  
S db N dup (?)
```

Дан непустой текст (последовательность из не более **N** символов, среди которых возможны малые латинские буквы), признак конца - точка. Требуется ввести с клавиатуры данный текст и записать его в начало рабочего массива **S**. Далее для введённого текста ответить на вопрос: **какая из малых латинских букв, входящих в состав текста, чаще всего встречается в нём (если таких букв несколько – напечатать любую из них)**.

Требования.

Для ввода текста воспользоваться функцией **ReadText(S)**, которая вводит текст (до точки) и размещает его в начале рабочего массива **S** (параметр функции – адрес рабочего массива). Функция возвращает (через **EAX**) длину введённого текста (без точки).

Для ответа на основной вопрос описать функцию **MaxLet(S,N)**, где **S** – адрес символьного массива, **N** – его размерность. Функция определяет, какая из малых латинских букв чаще всего встречается в этом массиве (если таких букв несколько, то выдать любую). Найденную букву вернуть через регистр **AL**.

**Обязательно:** в своей работе функция должна использовать вспомогательный (локальный) массив, отведя ему место в стеке (это массив из 26 байтов, в каждом байте которого хранится информация по числу вхождений соответствующей буквы латинского алфавита). Так как количество байтов, выделенное под область локальных переменных, должно быть кратно четырём, то размер этой области составит 28 байтов, адрес начала области [EBP-28]. Потребуется *три следующих этапа*: 1) инициализация нулями элементов локального массива; 2) просмотр массива символов (**S**) и корректировка элементов вспомогательного массива; 3) просмотр вспомогательного массива с целью поиска индекса максимального элемента. Тогда искомая буква – это найденный индекс плюс код буквы 'a'. Считать, что в используемой кодировке малые латинские буквы следуют строго друг за другом (в соответствии с алфавитом и без пропусков).

Обе функции (**ReadText** и **MaxLet**) должны выполнять стандартные соглашения о связях.

#### Задача 2. Матрицы (прямоугольные и квадратные)

(обязательно надо решить только пункты 1 и 4, пункты 2,3,5 – за дополнительные очки)

Описать в программе следующие данные:

```
.data  
MaxSize equ 150 ; предельный размер рабочего массива (для хранения матрицы).  
n db ? ; количество строк матрицы  
m db ? ; количество столбцов матрицы  
X dw MaxSize dup(?) ; рабочий массив для хранения матрицы размером n x m (где n*m ≤ MaxSize)  
; элементы матрицы рассматриваются как числа со знаком
```

Ввод элементов матрицы в память:

Программа запрашивает у пользователя параметры матрицы, с которой он желает работать: **n** - количество её строк (должно быть больше единицы), и **m** - количество её столбцов (должно быть больше единицы). Ввод осуществляется в переменные **n** и **m** по макрокомандам **inint n** и **inint m**. Если для введённых пользователем значений **n** и **m** не выполняется требование  **$n*m \leq \text{MaxSize}$**  (или какая-либо размерность равна единице), то программа выдаёт на экран соответствующее сообщение и далее спрашивает пользователя, желает ли он ввести другие параметры (Y/N). Если *нет* – то программа на этом завершает свою работу. Если *да* – программа ожидает ввода новых значений для **n** и **m** (т.е. новый шаг цикла). В случае задания корректных значений для **n** и **m** начинается *ввод элементов матрицы* (вводить элементы лучше *по строкам*, разделяя их друг от друга пробелами, нажимая ENTER после ввода каждой строки – это нормально, т.к. по правилам макрокоманды **inint op** вводимые числа как раз должны отделяться друг от друга либо “пробелами”, либо “концами строк”). В случае ввода лишних элементов – они отбрасываются (с помощью макрокоманды **flush**). Вводимые с клавиатуры элементы сохраняются в памяти в рабочем массиве **X**. *Эту часть задачи сделать без использования каких-либо процедур.*

После ввода элементов матрицы в память:

Программа *распечатывает* введённую матрицу *по строкам* (для реализации этой подзадачи предусмотреть соответствующую процедуру, параметры в которую следует передавать через стек – в соответствии со стандартными соглашениями о связях). *Требование:* в теле процедуры нельзя обращаться по имени к объектам, описанным вне процедуры (т.е. к именам **X**, **n** и **m**)  $\Rightarrow$  в качестве *параметров* в процедуру следует передавать адрес обрабатываемого массива и две его размерности.

Далее программа анализирует введённую матрицу и выдаёт на экран следующую информацию:

1) (обязательный пункт)

номера (через пробелы) тех строк, элементы которых упорядочены *по неубыванию* ( $\leq$ );

2) (дополнительный пункт за 5 очков)

номера (через пробелы) всех *симметричных* строк матрицы;

3) (дополнительный пункт за 5 очков)

номера (через пробелы) тех столбцов, все элементы в которых *одинаковые*;

*Замечание (важное !!!).*

Считать, что пользователь нумерует строки и столбцы, начиная с единицы, – это *логическая* (т.е. с точки зрения пользователя) нумерация строк и столбцов. Согласно такой нумерации и следует выдавать ответ на экран.

Если матрица оказалась квадратной (т.е. **n=m**), то программа дополнительно распечатывает в отдельных строках (через пробелы) следующую информацию:

4) (обязательный пункт)

все элементы, находящиеся на главной диагонали (начиная с элемента верхней строки);

5) (дополнительный пункт за 5 очков)

все элементы, находящиеся на побочной диагонали (начиная с элемента верхней строки);

(Для реализации каждого из пунктов 1)-5) следует предусмотреть свою процедуру (*требования* к процедурам аналогичны вышеописанным для *процедуры печати матрицы* – выполнять стандартные соглашения о связях).

После вывода требуемой (см. выше) информации программа запрашивает у пользователя, желает ли он работать с новой матрицей (Y/N)? Если нет, то на этом программа завершает свою работу. Если да, то экран очищается, и описанный выше процесс повторяется для следующей матрицы (запрашиваются новые значения для **n** и **m** и т.п.).