

БЛОК-4 (часть 2)

«Процедуры рекурсивные»

Основные правила при реализации рекурсии (помимо уже известных с 1-го семестра):

1) В обязательном порядке при входе в процедуру сохранять значения всех регистров, которые будут “портиться” в теле процедуры (кроме регистров, через которые возвращаются результаты), а перед выходом из процедуры – их восстанавливать. 2) Для спасения регистров использовать только стек.

Задача 3. « Системы счисления » (обязательная)

(нерекурсивная и рекурсивная процедуры; передача параметров через регистры)

Ввести (по макрокоманде **inint**) целое беззнаковое число (оно укладывается в 32-битовый формат). Затем ввести еще одно число (по макрокоманде **inint**) - основание системы счисления (от 2 до 10). Вывести на экран заданное число в виде цифр выбранной системы счисления.

Требования к решению.

В программе следует описать две процедуры (**OUTD1** и **OUTD2**), каждая из которых печатает значение регистра **EAX** как беззнаковое число в системе счисления, основание которой (от 2 до 10) передается через регистр **BL**. Отличие процедур состоит в том, что **OUTD1** – нерекурсивная, а **OUTD2** – рекурсивная. Необходимо последовательно применить к исходным данным (которые будут заданы на **EAX** и **BL**) сначала нерекурсивную процедуру, затем – рекурсивную. Ответы должны получиться одинаковые.

Обязательно: в каждой из процедур предусмотреть сохранение (в начале работы) и восстановление (перед окончанием работы) всех регистров, значения которых будут портиться в теле процедуры.

Внимание.

При реализации процедур обратить внимание на возможность переполнения в команде **DIV**.

Рекомендации по реализации процедур.

Нерекурсивная OUTD1.

Решать по аналогии с тем, как мы решали на семинаре пример на вывод числа в 5-ичной системе счисления (помещая в стек с помощью команды **push** все пятеричные цифры - 1 этап, извлекая из стека с помощью команды **pop** найденные цифры и печатая их на экран - 2 этап).

Рекурсивная OUTD2.

Простейший случай (величина в **EAX** меньше величины нового основания): значит в искомой записи числа всего одна цифра, выводим эту цифру (то есть содержимое **EAX**), завершаем работу процедуры.

Общий случай (величина в **EAX** больше или равна величины нового основания): значит в искомой записи числа содержится более одной цифры, готовимся к рекурсивному вызову (делим число на новое основание, вызываем рекурсивно **OUTD2** для аналогичной обработки целой части от деления, затем печатаем остаток, завершаем работу процедуры).

Задача 4 . « Текст наоборот » (рекурсивная процедура без параметров) (дополнительная за 10 очков)

Ввести (по макрокоманде **inchar**) последовательность символов (*отличных от точки*), за которой следует точка. Напечатать эту же последовательность, но **в обратном порядке** (*без точки*).

Требование к решению.

Описать в программе вспомогательную рекурсивную процедуру **REVERSE** без параметров, которая вводит последовательность символов и печатает ее в обратном порядке.

Обязательно: в процедуре предусмотреть сохранение (в начале работы) и восстановление (перед окончанием работы) всех регистров, значения которых будут портиться в теле процедуры (иначе рекурсия работать не будет!).

Рекомендации по реализации процедуры.

Начало: ввод символа, сравнение его с точкой.

Простейший случай (точка): текст закончился, завершаем работу процедуры.

Общий случай (не точка): вызываем рекурсивно **REVERSE** для аналогичной обработки оставшейся части текста, после чего печатаем прочитанный в *начале* символ (ведь первый прочитанный символ должен быть напечатан в последнюю очередь!), завершаем работу процедуры.

Задача 5 . « Отрицательные и положительные числа » (рекурсивная процедура без параметров) (дополнительная за 10 очков)

Ввести (по макрокоманде **inint**) последовательность ненулевых чисел, за которой следует 0. Вывести сначала все **отрицательные числа**, а затем – все **положительные** (в любом порядке) этой последовательности.

Требование к решению.

Описать в программе вспомогательную рекурсивную процедуру **PRINT** без параметров, которая вводит эти числа и выводит сначала все отрицательные числа, а затем – все положительные (в любом порядке).

Обязательно: в процедуре предусмотреть сохранение (в начале работы) и восстановление (перед окончанием работы) всех регистров, значения которых будут портиться в теле процедуры (иначе рекурсия работать не будет!).

Рекомендации по реализации процедуры.

Начало: ввод числа, сравнение его с нулём.

Простейший случай (ноль): последовательность закончилась, завершаем работу процедуры.

Общий случай (не ноль): если число **отрицательное**, то печатаем его, вызываем рекурсивно **PRINT** для аналогичной обработки оставшейся части последовательности, завершаем работу процедуры; если число **положительное**, то сначала вызываем рекурсивно **PRINT** для аналогичной обработки оставшейся части последовательности, затем печатаем (*положительное*) число, завершаем работу процедуры.

Задача 6. Максимальная цифра в десятичной записи числа (обязательная)

Ввести (по макрокоманде **inint**) число без знака (число укладывается в 32-битовый формат), вывести максимальную цифру в десятичной записи этого числа.

Например: 283709234 → 9

Требование. В программе описать **рекурсивную** функцию **MaxDig(N)** со стандартными соглашениями о связях, где **N** – двойное слово без знака (передается по значению). Функция возвращает (через **EAX**) максимальную цифру в десятичной записи числа **N**.

Задача 7. Наибольший общий делитель (дополнительная за 10 очков)

Ввести (по макрокоманде **inint**) два числа без знака (числа укладываются в 32-битовый формат), вывести их наибольший общий делитель.

Например: для чисел 1 357 966 350 и 2 456 789 600 ответ (НОД) равен 50.

Требование. В программе описать **рекурсивную** функцию **GCD(A, B)** со стандартными соглашениями о связях, где **A** и **B** – двойные слова без знака (передаются по значению). Функция возвращает (через **EAX**) значение наибольшего общего делителя этих двух чисел.

Задача 8. Максимум и минимум (дополнительная за 10 очков)

Для ввода задана последовательность символов, представляющая собой правильную запись формулы следующего вида:

<формула> ::= цифра | **M**(<формула>, <формула>) | **m**(<формула>, <формула>)

<цифра> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

M трактуется как максимум (max), а **m** - как минимум (min).

Описать рекурсивную процедуру **MaxMin** без параметров, которая вводит эту формулу, вычисляет ее значение (как число) и присваивает его регистру **AL**.

Требование. Описать в программе **рекурсивную** функцию **MaxMin** (без параметров), которая вводит эту формулу, вычисляет ее значение (как число) и возвращает ответ через регистр **AL**.

Задача 9. Формула (дополнительная за 10 очков)

Описать программу, которая вводит текст вида

<формула> = <формула> , где

<формула> ::= <цифра> | (<формула> <знак> <формула>)

<знак> ::= + | - | *

<цифра> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

и которая определяет, равны ли значения двух указанных формул, выводя на экран ответ **ДА** или **НЕТ**.

Требование. В программе описать и использовать **рекурсивную** функцию (без параметров) **Form**, которая вводит и вычисляет значение **одной** формулы, возвращая ответ через регистр **EAX**.