

Контрольная работа (на машинах) №4

Все используемые здесь процедуры и функции должны соблюдать стандартные соглашения о связях stdcall !

Задача (состоит из *четырёх этапов*, начало выполнения – обязательно *1-ый этап* !)

Общее пожелание: везде, где это уместно, рекомендуется использовать при работе с записями операторы **mask**, **width** (при задании величины сдвигов можно использовать также значения имён полей).

Date_pack record D:5, M:4, Y:7 ; упакованная дата в формате «день-месяц-год»

; (хранит значение даты в рамках некоторого столетия)

; в поле **D** хранится номер дня (в месяце): число от 1 до 31

; в поле **M** хранится номер месяца: число от 1 до 12

; в поле **Y** хранятся две последние цифры года: число от 0 до 99

; считать, что ДАТА ЗАДАНА КОРРЕКТНО (т.е. значения для **D**, **M** и **Y** – в рамках указанных диапазонов)

Этап 1 (выполняется в первую очередь) (30 очков)

Описать переменные-записи **D1** и **D2** типа **Date_pack**. Ввести с клавиатуры значения для **D1** и **D2** (считать, что ввод дат будет выполнен корректно). Проверить, предшествует ли дата **D1** дате **D2** (в рамках некоторого столетия), т.е. проверить условие **D1<D2** ? Вывести ответ в виде: «условие **D1<D2** истинно/ложно» (где вместо **D1** и **D2** напечатать значения дат в формате **dd.mm.yy**).

Требования к решению на 1-ом этапе.

1) Описать процедуру **In_Rec(D)**, которая вводит (с клавиатуры) тройку чисел – значения полей записи типа **Date_pack** и присваивает введенную запись параметру **D** (внимание: передача параметра должна осуществляться по ссылке !).

Замечание: макрокоманда **inint** считает “концом” вводимого числа первую нецифру, которая “проглатывается”; поэтому при вводе тройку чисел можно задавать либо в виде **3/05/18**, либо в виде **3.5.18** (либо отделять цифры пробелом и т.п.), осуществляя ввод этих трёх чисел путём последовательного выполнения трёх макрокоманд **inint**.

2) Описать также функцию **Less(D1,D2)**, которая проверяет условие **D1<D2** (где **D1** и **D2** – записи типа **Date_pack**, переданные в функцию по значению) и возвращает (через **AL**) ответ **1** (да) или **0** (нет).

3) Описать процедуру **Out_Rec(D)** для вывода на экран значения даты-записи **D** в формате **dd.mm.yy** (параметр **D** передать в процедуру по значению).

Этап 2 (выполняется только после сдачи этапа 1) (30 очков)

N equ 6

Arr_of_Rec Date_pack N dup(<>); массив с именем **Arr_of_Rec** из **N** элементов-дат типа **Date_pack**.

Ввести (с клавиатуры) даты, сохранив их в последовательных элементах массива **Arr_of_Rec**.

Распечатать получившийся массив дат. Далее проверить «упорядочены ли введенные даты по возрастанию?» и выдать ответ «введенные даты упорядочены/не упорядочены».

Если даты упорядочены по возрастанию, то перейти к *этапу 4*. Иначе – перейти к *этапу 3*.

Требования к решению на 2-ом этапе.

Для ввода, вывода и сравнения дат использовать ранее (на 1-ом этапе) отлаженные процедуры **In_Rec(D)**, **Out_Rec(D)** и функцию **Less(D1,D2)**.

Других процедур и функций на этом этапе не вводить, то есть в основной программе следует реализовать *три последовательных цикла*:

1) цикл для ввода **N** дат и сохранения их в элементах массива **Arr_of_Rec**;

2) цикл для вывода **N** дат, хранящихся в элементах массива **Arr_of_Rec**;

3) цикл для проверки «упорядочены ли введенные элементы-даты по возрастанию?».

----- *продолжение смотрите на следующей странице* -----

Этап 3 (выполняется только после сдачи *этапа 2*) (30 очков)

Найти среди элементов-дат массива **Arr_of_Rec** наименьшую дату (т.е. предшествующую всем остальным датам) и вывести её на экран (описание массива **Arr_of_Rec** дано на *этапе 2*). На этом завершить работу программы.

Требования к решению на 3-ом этапе.

- 1) Для поиска в массиве **Arr_of_Rec** наименьшей даты реализовать функцию **Min_Date(Arr,Len)**, где **Arr** – адрес массива из записей-дат, **Len** – его длина. Функция возвращает через регистр **AX** минимальную дату. В процессе своей работы функция для сравнения текущего минимума с очередной датой обращается к ранее отлаженной функции **Less(D1,D2)**.
- 2) Для вывода наименьшей даты использовать ранее отлаженную процедуру **Out_Rec(D)** .

Этап 4 (выполняется после сдачи *этапа 2*) (30 очков)

Date_unpack struc ; распакованная дата

D db ? ; день (число от 1 до 31)

M db ? ; месяц (число от 1 до 12)

Y db ? ; год (число от 0 до 99)

Date_unpack ends

; **N equ 6** – эта константа должна быть описана выше (для 2-го этапа)

Arr_of_Struc Date_unpack N dup(<>); массив с именем **Arr_of_Struc** из **N** элементов типа **Date_unpack**

Распаковать элементы-даты массива **Arr_of_Rec** (см. *этап 2*), сохранив их в массиве **Arr_of_Struc**. Вывести на экран получившийся массив **Arr_of_Struc** (каждую дату - в формате **dd.mm.yy**). На этом завершить работу программы.

Требования к решению на 4-ом этапе.

Для распаковки даты описать процедуру **Rec_to_Struc(D_pack,D_unpack)**, которая по записи типа **Date_pack** формирует структуру типа **Date_unpack** (1-ый параметр – по значению, 2-ой параметр – по ссылке).

Для вывода полей структуры описать процедуру **Out_Struc(D_unpack)** (параметр в процедуру передать по ссылке).

Внимание! В теле процедур **Rec_to_Struc** и **Out_Struc** при обращении к полям структуры настоятельно рекомендуется использовать имена этих полей. Для этого следует применить к адресу структуры оператор **Date_unpack ptr**. Например, если в регистре **EBX** хранится адрес структуры типа **Date_unpack**, то для доступа к полю **M** этой структуры нужно использовать конструкцию **(Date_unpack ptr [EBX]).M** (такая конструкция задаёт адрес, тип которого совпадёт с типом поля **M**, то есть равен байту).

Других процедур и функций на этом этапе не вводить, то есть для *этапа 4* требуется в основной программе реализовать два последовательных цикла:

- 1) цикл движения по массиву **Arr_of_Rec** с целью распаковки дат и записи их в массив **Arr_of_Struc**;
- 2) цикл движения по массиву **Arr_of_Struc** с целью вывода дат на экран (каждую дату - в формате **dd.mm.yy**).

Этап 5 (дополнительный, выполняется только после сдачи *всех четырёх этапов*) (30 очков)

Вместо функции **Min_Date(Arr,Len)** (см. *этап 3*), ввести специально для *этапа 5* новую функцию **Ptr_to_Min_Date(Arr,Len)**. Эта функция должна возвращать через **EAX** адрес (смещение) минимального элемента массива **Arr** длины **Len** (элементы массива - типа **Date_pack**). Воспользоваться новой функцией для сортировки элементов массива **Arr_of_Rec** методом *выбора* (*догадаться самостоятельно - как нужно применить в решении новую функцию*).