МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФГБОУ ВО «ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ И.С.ТУРГЕНЕВА»

Кафедра «Программная инженерия»

**Отчет**

По лабораторной работе №6  
на тему:  
“Обработка структур. Введение базы данных о пациентах”  
по дисциплине: “Архитектура ЭВМ и систем”

Работу выполнил:  
Студент группы 01-ИТ

Колчев Д.Н.

Проверил:  
Доцент кафедры  
программной инженерии

Конюхова О.В.

Орёл, 2020

1. **Структура программы**

;Зарезервируем в Stack 20 эл-в

s\_s segment

db 100 dup('0')

s\_s ends

Person STRUC

;Карта, пол, год рождения, даты нач/конц лечения.

nCard dw 0

sex db ?

bYear dw 0

dtAd db ' / / '

dtLv db ' / / '

Person ENDS

d\_s SEGMENT

;Определяем данные 3-х пациентов.

pt1 Person <44,'m',1990, '10/10/19','30/10/19'>

pt2 Person <45,'m',1990, '10/10/19','31/10/19'>

pt3 Person <46,'f',1995, '23/10/19','31/10/19'>

;Конкретные даты нач/конц лечения для поиска.

dtAdSrch db '10/10/19', '$'

dtLvSrch db '31/10/19', '$'

dtBYearSrch dw 1990, '$'

nCardSrch dw 45, '$'

;Результаты выполнения программы:

task1 db 0

task2 db 0

task3 dw 0

task4 db 0

d\_s ENDS

c\_s SEGMENT

ASSUME ds:d\_s,cs:c\_s,ss:s\_s

;Скрыть внешний цикл

pushcx PROC NEAR

RET

pushcx ENDP

BEGIN: ; начало программы

mov ax,d\_s

mov ds,ax

xor ax,ax

;1)Количество пациентов поступивших 10/10/19?

;Внешний цикл для 3-х пациентов (pt1-pt2) -> call

;Внутренний цикл для проверки 8 символов даты

;Передаём в al данные о 1-м пациенте

lea di, pt1

mov cx, 3

nxtPt:

push cx

;Передаём дату пациента

mov cx, 8

lea si, [di].dtAd

rdNxt:

lodsb

mov bl, al

xor ax, ax

push si

;Кастыль для работы со второй датой

cmp dl, 0

jz cont ;=

mov si, bp

jmp more

;Передаём дату для поиска

cont:

lea si, dtAdSrch

more:

lodsb

mov bp, si ;Сохраняем новую si

;Сравниваем полученные даты

cmp al, bl

jz same

jmp difer

;Даты совпали - +1 совпадение

same:

inc dl

;Если совпали все 8 цифр

cmp dl, 8

je yes

jmp no

;Мы нашли искомого пациента, ищем след.

yes:

inc dh ;Пациентов +1 совпало

jmp difer

;Мы не нашли пациета, читаем след. запись

no:

jmp difer

difer:

pop si

Loop rdNxt

;Проверяем дату следующего пациента

pop cx

add di, type Person

xor dl, dl

Loop nxtPt

mov task1, dh

;2)Количество женщин, выписанных 31/10/19?

;Внешний цикл для 3-х пациентов (pt1-pt2) -> call

;Внутренний цикл для проверки поля 'sex'

;Нулируем регистры

xor ax, ax

xor bx, bx

xor dx, dx

;Передаём в al данные о 1-м пациенте

lea di, pt1

mov cx, 3

nxtPt1:

push cx

;Передаём дату выписки пациента

mov cx, 8

lea si, [di].dtLv

rdNxt1:

lodsb

mov bl, al

xor ax, ax

push si

;А может это и не костыль, а часть проги?

cmp dl, 0

jz cont1

mov si, bp

jmp more1

;Передаём даты выписки для поиска

cont1:

lea si, dtLvSrch

more1:

lodsb

mov bp, si

;Сравниваем полученные даты

cmp al, bl

jz same1

jmp difer1

;Цифры дат совпали - +1 совпадения

same1:

inc dl

;Если совпали все 8 цифр

cmp dl, 8

je wom1

jmp no1

;Проверяем, является ли пациент женщиной

wom1:

mov bl, 'f'

mov al,[di].sex

cmp bl, al

je yes1

jmp no1

;Мы нашли искомого пациента, ищем след.

yes1:

inc dh ;Пациентов +1 совпало

jmp difer1

;Мы не нашли пациета, читаем след. запись

no1:

jmp difer1

difer1:

pop si

Loop rdNxt1

;Проверяем дату следующего пациента

pop cx

add di, type Person

xor dl, dl

Loop nxtPt1

mov task2, dh

;3)Год рождения пациента с номером карты (45)?

;Внешний цикл для 3-х пациентов (pt1-pt2)

;Внутренний цикл для проверки номера карты

;Нулируем регистры

xor ax, ax

xor bx, bx

xor dx, dx

;Передаём в al данные о 1-м пациенте

lea di, pt1

mov cx, 3

nxtPt2:

push cx

;Передаём дату выписки пациента

mov cx, 2

lea si, [di].nCard

rdNxt2:

lodsb

mov bl, al

xor ax, ax

push si

;Не кастыль, а фича!

cmp dl, 0

jz cont2

mov si, bp

jmp more2

;Передаём даты выписки для поиска

cont2:

lea si, nCardSrch

more2:

lodsb

mov bp, si

;Сравниваем полученные даты

cmp al, bl

jz same2

jmp difer2

;Цифры дат совпали - +1 совпадения

same2:

inc dl

;Если совпали все 2 цифры

cmp dl, 2

je year2

jmp no2

;Проверяем, является ли пациент женщиной

year2:

mov bl, 45

mov ax,[di].nCard

cmp bl, al

je yes2

jmp no2

;Мы нашли искомого пациента, ищем след.

yes2:

inc dh ;Пациентов +1 совпало

;Записываем год рождения

lea si, [di].bYear

lodsw

mov task3, ax

;Мы не нашли пациета, читаем след. запись

no2:

jmp difer2

difer2:

pop si

Loop rdNxt2

;Проверяем дату следующего пациента

pop cx

add di, type Person

xor dl, dl

Loop nxtPt2

;4)Количество мужчин 1990 года рождения?

;Внешний цикл для 3-х пациентов (pt1-pt2)

;Внутренний цикл для проверки года рождения и поля 'sex'

;Нулируем регистры

xor ax, ax

xor bx, bx

xor dx, dx

;Передаём в al данные о 1-м пациенте

lea di, pt1

mov cx, 3

nxtPt3:

push cx

;Передаём дату выписки пациента

mov cx, 2

lea si, [di].bYear

rdNxt3:

lodsb

mov bl, al

xor ax, ax

push si

;Не кастыль, а фича!

cmp dl, 0

jz cont3

mov si, bp

jmp more3

;Передаём даты выписки для поиска

cont3:

lea si, dtBYearSrch

more3:

lodsb

mov bp, si

;Сравниваем полученные даты

cmp al, bl

jz same3

jmp difer3

;Цифры дат совпали - +1 совпадения

same3:

inc dl

;Если совпали все 8 цифр

cmp dl, 2

je man3

jmp no3

;Проверяем, является ли пациент женщиной

man3:

mov bl, 'm'

mov al,[di].sex

cmp bl, al

je yes3

jmp no3

;Мы нашли искомого пациента, ищем след.

yes3:

inc dh ;Пациентов +1 совпало

jmp difer3

;Мы не нашли пациета, читаем след. запись

no3:

jmp difer3

difer3:

pop si

Loop rdNxt3

;Проверяем дату следующего пациента

pop cx

add di, type Person

xor dl, dl

Loop nxtPt3

mov task4, dh

mov ah,4ch

int 21h

c\_s ends

end begin

1. **Контрольные вопросы:**
2. **Структуры и определение шаблона структуры в программе.**

Cтруктура — это тип данных, состоящий из фиксированного числа элементов разного типа.  
 Шаблон структуры:  
 имя\_структуры STRUC  
 <описание полей>  
 имя\_структуры ENDS  
Местоположение шаблона в программе может быть произвольным, но, следуя логике работы однопроходного транслятора, он должен быть расположен до того места, где определяется переменная с типом данной структуры. То есть при описании в сегменте данных переменной с типом некоторой структуры ее шаблон необходимо поместить в начале сегмента данных либо перед ним.

1. **Инициализация полей структуры в программе.**  <описание полей> представляет собой последовательность директив описания данных db, dw, dd, dq и dt.  
    Их операнды определяют размер полей и, при необходимости, начальные значения. Этими значениями будут, возможно, инициализироваться соответствующие поля при определении структуры.  
    Для использования описанной с помощью шаблона структуры в программе необходимо определить переменную с типом данной структуры. Для этого используется следующая синтаксическая конструкция:  
    [имя переменной] имя\_структуры <[список значений]>  
    имя переменной – идентификатор переменной данного структурного типа.  
    Задание имени переменной необязательно. Если его не указать, будет просто выделена область памяти размером в сумму длин всех элементов структуры.  
    список значений – заключенный в угловые скобки список начальных значений элементов структуры, разделенных запятыми. Его задание также необязательно. Если список указан не полностью, то все поля структуры для данной переменной инициализируются значениями из шаблона, если таковые заданы.
2. **Режимы адресации для доступа к элементам структуры, для доступа к элементам массива структуру.** 1. Базовый. Применяется для доступа к структурам данных переменной длины. Тогда базовый адрес, определяющий начало набора элементов, хранится в базовом регистре, а смещение в команде определяет расстояние до определённого элемента.  
    EA = {BP/BX} + disp  
    Этот режим адресации удобно использовать для записей – структур данных, содержащих поля различной длины и, возможно, различных типов.

2. Базово-индексный. Используется для доступа к элементам массива, адресуемого указателем. Базовый адрес массива задаётся указателем базы (базовым регистром), а номер элемента массива – содержимым индексного регистра.  
 EA = {BP/BX} + {SI/DI}  
 Если в bx содержится 100, а в si находится 52, то по адресу (смещению) 152 в сегменте данных находится искомое данное.  
 Такой режим адресации удобно использовать при работе со сложными структурами данных, поскольку он позволяет изменять две адресные компоненты.  
 3. Относительный базовый индексный. Используется для адресации элементов в указываемом массиве записей. Базовый адрес массива задаётся указателем базы, номер записи (т.е., элемента массива) определяется содержимым индексного регистра, а смещение в команде указывает расстояние до записи.  
 EA = {BP/BX} + {SI/DI} + disp

1. **Каким образом вычисляется расстояние до некоторого поля отдельного элемента массива структур?** Идея введения структурного типа в любой язык программирования состоит в объединении разнотипных переменных в один объект.  
    В языке должны быть средства доступа к этим переменным внутри конкретного экземпляра структуры. Для того, чтобы сослаться в команде на поле некоторой структуры, используется специальный оператор - символ "." (точка). Он используется в следующей синтаксической конструкции:  
    адресное\_выражение.имя\_поле\_структуры  
    Адресное\_выражение - идентификатор переменной некоторого структурного типа или выражение в скобках в соответствии с указанными ниже синтаксическими правилами;  
    Имя\_поле\_структуры - имя поля из шаблона структуры  
    Это,на самом деле, тоже адрес, а точнее, смещение поля от начала структуры.  
    Таким образом оператор "." (точка) вычисляет выражение  
    (адресное\_выражение) + (имя\_поля\_структуры)
2. **Назначение оператора TYPE.** Поле TYPE доопределяет тип дескриптора. Проверка типа позволяет обнаруживать ошибки при использовании сегмента не по назначению. Например, в сегмент кода нельзя записывать, а с сегментом таблицы LDT нельзя явно производить операции считывания, записи и выполнения.  
    Права, которыми может быть помечен несистемный дескриптор, ограничивают множество операций, которые можно производить с его сегментом. Права указываются также в поле TYPE несистемных дескрипторов. Например, исполняемый сегмент кода дополнительно может быть помечен как читаемый.