БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра программного обеспечения информационных технологий

Факультет ИНО

Специальность ПОИТ

Индивидуальная практическая работа №2

по дисциплине «Языки программирования»

часть 1

Вариант № 12

Выполнила: Н. Г. Карпеко

Обучение по сертификатам

Минск 2020

**Тема работы. Массивы.**

Задан одномерный массив. Найти сумму его элементов. Результат вывести на экран, вставив нужные цифры в шаблон текста сообщения.

Например, "сумма элементов массива 50". Массив задать в сегменте данных или ввести с клавиатуры.

Файл ipr212.asm содержит исходный код программы с реализацией задания. Компиляция, сборка и запуск программы выполнены в DOSBox Ver. 0.74-3. Массив из 10 однобайтных чисел задан в сегменте данных. Сумма элементов массива загружается в аккумулятор ах. Сумма выводится на экране после реализации вывода сообщения 'Summa massiva = $'.

Листинг программы:

stk segment stack ; сегмент стека

db 256 dup (?) ; под стек отвели 256 байт

stk ends ; конец сегмента стека

;-----------

data segment ; сегмент данных

mas db 32, 7, 5, 8, 4, 3, 0, 27, 11, 9 ; объявление исходного массива

len dw $-mas ;размер массива в байтах

Mess db 'Summa massiva = $' ; вывод на экран сообщения

data ends ; конец сегмента данных

; ------------

code segment ; открыть сегмент кода

start: ; метка начала программы

assume cs:code, ds: data, ss:stk ; assume говорит какой сегментный

; регистр использовать для адресации внутри сегмента

mov ax, data ; в ах загрузили данные из сегмента данных

mov ds, ax ; загрузить сегментный регистр данных DS

lea bx, mas ; берём в bx адрес первого элемента

mov cx, len ; счетчик цикла, в len - количество элементов

**xor** ax, ax ; обнулили ах: в ах - 0

cikl: add al, [bx] ;прибавляем к al байт, адрес которого хранится в bx

inc bx ; переходим к следующему элементу

**loop** cikl ; прошли 1 цикл, берём следующий элемент массива

aam ; преобразуем сумму в BCD-код. коррекция неупакованного

; BCD-умножения

add ax, 3030h ; затем преобразуем ее в ASCII-код

;--------------

mov bx, ax ; начинаем вывод сообщения и суммы на экран

mov ah, 09h ; в регистре ah хранят функцию вывода строки (09h)

mov dx, offset Mess ; в dx поместили смещение Messg (строки сообщения)

; для ее вывода на экран

int 21h ; вызываем прерывание ОС DOS, идёт вывод строки текста

mov ah, 02h ; выводим полученное число (сумму элементов массива)

mov dl, bh

int 21h ; прерывание для вывода строки

mov dl, bl

int 21h ; прерывание

;------------

mov ax, 4c00h ; функция 4ch означает завершение работы программы и

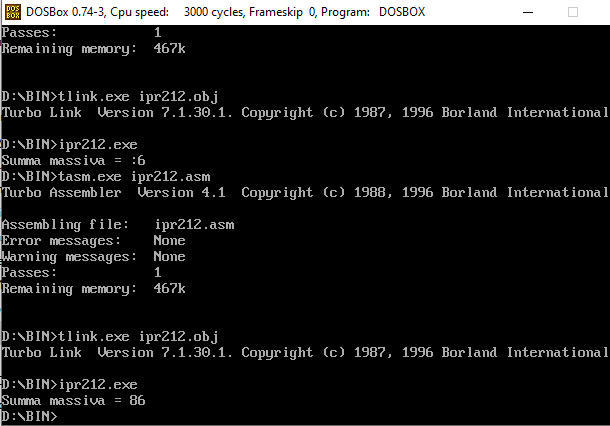
; возврат в систему DOS (корректный выход из программы в DOS)

int 21h ; вызов DOS

code ends ; конец сегмента кода

**end** start ; выход из программы (закрыть программу).

Результат запуска программы в DOSBox:



Ответ: сумма элементов массива 32, 7, 5, 8, 4, 3, 0, 27, 11, 9 равна 86.