МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №5**

**з дисципліни «Архітектура комп'ютера»**

## на тему:« Реалізація циклічних алгоритмів мовою Асемблера.»

Виконав: студент гр. ПЗ2011

Кулик Сергій Вадимович

Прийняла: ас. каф. КІТ

Нежуміра О. І.

Дніпро, 2020

Тема. Реалізація циклічних алгоритмів мовою Асемблера.

Мета. Набути практичних навичок розробки програм циклічної структури.

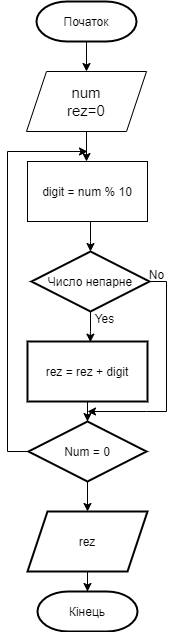
## Завдання

Розробити алгоритм та написати програму за індивідуальним завданням. Вхідні дані задаються у сегменті даних. Результати роботи програми зберігаються у сегменті даних.

Індивідуальне завдання

Задано натуральне число, діапазон значень – слово. Визначити суму непарних цифр числа.

1. Алгоритм (блок-схема)

****

1. Лістинг

Turbo Assembler Version 3.0 11/18/20 13:22:27 Page 1

ak55.asm

Turbo Assembler Version 3.0 11/18/20 13:22:27 Page 1

ak55.asm

1 ;=======================================================================================#

2 ;Розробити алгоритм та написати програму за +

3 індивідуальним завданням (табл.2). #

4 ;Вхідні дані задаються у сегменті даних. +

5 #

6 ;Результати роботи програми зберігаються у +

7 сегменті даних. #

8 ;Задано натуральне число, діапазон значень – слово. +

9 Визначити суму непарних цифр числа. #

10 ;=======================================================================================#

11 ;сегмент стека

12 0000 stk segment stack

13 0000 80\*(??) db 128 dup(?)

14 0080 stk ends

15

16 ;сегмент даних

17 0000 data segment para public 'data'

18 0000 350B num dw 13579

19 0002 0000 rez dw 0

20 0004 data ends

21 ;сегмент команд

22 0000 code segment para public 'code'

23 assume cs:code, ds:data, ss:stk

24 0000 begin:

25 0000 B8 0000s mov ax, data

26 0003 8E D8 mov ds, ax

27 ;=============================

28 0005 A1 0000r mov ax, num

29 0008 BB 000A mov bx, 10

30 000B 33 C9 xor cx, cx

31 000D while\_begin: ;Початок циклу

32 000D 33 D2 xor dx, dx ;dx = 0

33 000F F7 F3 div bx ;Ділення на 10

34 0011 F7 C2 0001 test dx, 1 ;Перевірка на парність

35 0015 75 02 ~~jnz sum ;якщо цифра парна, перейти по мітці+~~

~~36 sum~~

37 0017 74 02 jz next ;якщо цифра непарна, перейти по +

38 мітці next

39 0019 ~~sum:~~

40 0019 03 CA add cx, dx

41 001B next:

42 001B A9 FFFF test ax, 0FFFFh ;Якщо число не дорівнюэ 0, перейти на +

43 початок циклу

44 001E 75 ED jnz while\_begin

45 ;=============================

46 0020 outnum: ;організація виводу числа в +

47 консоль

48 0020 89 0E 0002r mov rez, cx

49 0024 8B C1 mov ax, cx

50 0026 D4 0A aam

51 0028 05 3030 add ax,3030h

52 002B 8A D4 mov dl,ah

53 002D 8A F0 mov dh,al

54 002F B4 02 mov ah,02

55 0031 CD 21 int 21h

56 0033 8A D6 mov dl,dh

57 0035 CD 21 int 21h

\_Turbo Assembler Version 3.0 11/18/20 13:22:27 Page 2

ak55.asm

58 ;=============================

59 0037 endprog:

60 0037 B8 4C00 mov ax, 4C00h

61 003A CD 21 int 21h

62 003C code ends

63 end begin

\_Turbo Assembler Version 3.0 11/18/20 13:22:27 Page 3

Symbol Table

Symbol Name Type Value

??DATE Text "11/18/20"

??FILENAME Text "ak55 "

??TIME Text "13:22:27"

??VERSION Number 0300

@CPU Text 0101H

@CURSEG Text CODE

@FILENAME Text AK55

@WORDSIZE Text 2

BEGIN Near CODE:0000

ENDPROG Near CODE:0037

NEXT Near CODE:001B

NUM Word DATA:0000

OUTNUM Near CODE:0020

REZ Word DATA:0002

SUM Near CODE:0019

WHILE\_BEGIN Near CODE:000D

Groups & Segments Bit Size Align Combine Class

CODE 16 003C Para Public CODE

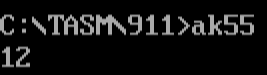
DATA 16 0004 Para Public DATA

STK 16 0080 Para Stack

1. Тести

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва | Вхідні дані | Очікувані результати |
| 1. | Довільне число | 27562 | 12 |
| 2. | Усі цифри числа парні | 24242 | 0 |
| 3. | Усі цифри числа непарні | 13579 | 25 |

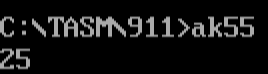
1. Результати тестувань
   1. Довільне число



* 1. Усі цифри числа парні



* 1. Усі цифри числа непарні



Усі результати роботи программи збігаються з очікуваними результатами тестів. Програма виводе в консоль сумму непарних цифр числа.

1. Висновок

Для реалізації циклу на мові ассемблер використовуються команди умовного та безумовного переходу. Для створення циклу з лічильником, необхідно у регістр СХ занести кількість ітерацій, а команда Loop буде змешувати лічильник. Цикл виконується тоди, доки лічильник не стане рівним 0