МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



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

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №7**

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

## на тему:« Організація введення/виведення інформації з консолі в мові Ассемблера. Обробка масивів чисел в мові Ассемблера..»

Виконав: студент гр. ПЗ2011

Кулик Сергій Вадимович

Прийняла: ас. каф. КІТ

Нежуміра О. І.

Дніпро, 2020

Тема. Організація введення та виведення інформації в консолі на рівні Ассемблера. Обробка масивів чисел в мові Ассемблера.

Мета. Вивчити засоби мови Ассемблера для введення та виведення числової інформації. Набути практичних навичок з обробки двомірних числових масивів мовою Ассемблера.

**1. Завдання**

**Рівень Е - D (достатньо - задовільно)**

Дано одновимірний масив. Елементи масиву – цілі числа зі знаком з діапазону слово. Довжина масиву від 5 до 10 елементів. Обробити масив згідно індивідуального завдання з таблиці 1.

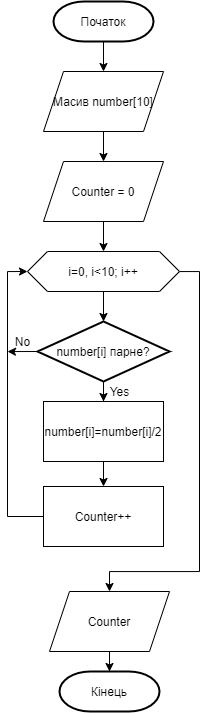
Варіант 5

Зменшити удвічі всі парні елементи і порахувати їх кількість.

**2. Тести**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва тесту | Вхідні дані | Результат |
| 1 | Усі числа парні | 2, 4, 6, 8, 2 | 1, 2, 3, 4, 1  Замін 5 |
| 2 | Усі числа непарні | 53, 21, 9, 17, 1 | 53, 21, 9, 17, 1  Замін 0 |
| 3 | Парні та непарні числа | 20, 42, 15, 31, 64 | 10, 21, 15, 31, 32,  Замін 3 |

**3. Алгоритм**

****

**4. Лістинг програми**

Turbo Assembler Version 3.0 12/21/20 17:37:13 Page 1

ak7.asm

1 ;==============================================================

2 ;Програма знаходе всі парні числа у масиві та +

3 зменшує удвічі їхнє значення

4 ;Елементи масиву – цілі числа зі знаком з діапазону +

5 слово.

6 ;Вхід: розмір масиву та його елементи вводяться з +

7 клавіатури.

8 ;Вихід: Кількість замін у масиві, оновлений масив +

9 виводиться на консоль.

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 0014 len dw 20 ;максимальний розмір масиву

19 0002 ???? counter dw ? ;розмір масиву

20 0004 14\*(????) A dw 20 dup (?) ;масив

21 002C 02 x db 2

22 002D ???? quantity dw ?

23 ;буфер для введення числа в ASCII-формі (максимум 6 +

24 символів (-32768) + кінець рядка (0dh))

25 002F 07 ?? 07\*(??) buffer db 7, ?, 7 dup (?)

26 0038 45 6E 74 65 72 20 74+ msgCounter db 'Enter the number of elements in the array N (N <= 20) : $'

27 68 65 20 6E 75 6D 62+

28 65 72 20 6F 66 20 65+

29 6C 65 6D 65 6E 74 73+

30 20 69 6E 20 74 68 65+

31 20 61 72 72 61 79 20+

32 4E 20 28 4E 20 3C 3D+

33 20 32 30 29 20 3A 20+

34 24

35 0071 45 6E 74 65 72 20 61+ msgArray db 'Enter array elements:', 0ah, 0dh, '$'

36 72 72 61 79 20 65 6C+

37 65 6D 65 6E 74 73 3A+

38 0A 0D 24

39 0089 41 20 5B 20 24 msgPrompt1 db 'A [ $'

40 008E 20 5D 20 3D 20 24 msgPrompt2 db ' ] = $'

41 0094 0A 0D 45 52 52 4F 52+ msgError db 0ah, 0dh, 'ERROR!', 0ah, 0dh, '$'

42 21 0A 0D 24

43 009F 41 72 72 61 79 20 3A+ msgPrint db 'Array :', 0ah, 0dh, '$'

44 0A 0D 24

45 00A9 50 72 65 73 73 20 61+ msgPause db 'Press any key ...', '$'

46 6E 79 20 6B 65 79 20+

47 2E 2E 2E 24

48 00BB 4E 75 6D 62 65 72 20+ msgQuantity db 'Number of replacements: ', '$'

49 6F 66 20 72 65 70 6C+

50 61 63 65 6D 65 6E 74+

51 73 3A 20 24

52 00D4 0A 0D 24 CrLf db 0ah, 0dh, '$'

53

54 00D7 data ends

55

56 ;сегмент команд

57 0000 code segment para public 'code'

Turbo Assembler Version 3.0 12/21/20 17:37:13 Page 2

ak7.asm

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

59 0000 begin:

60 0000 B8 0000s mov ax, data

61 0003 8E D8 mov ds, ax

62

63 ;введення кількості елементів масиву (цикл +

64 повторюється до введення "правильного" числа)

65 0005 mInCounter:

66 0005 BA 0038r lea dx, msgCounter

67 0008 E8 00E7 call outString

68 000B BA 002Fr lea dx, buffer

69 000E B4 0A mov ah, 0ah

70 0010 CD 21 int 21h

71 0012 80 3E 0030r 00 cmp buffer + 1, 0

72 0017 74 EC jz mInCounter

73 0019 E8 00F3 call convertStrToInt

74 001C 72 17 jc mErrorCounter

75 001E 3B 06 0000r cmp ax, len

76 0022 7F 11 jg mErrorCounter

77 0024 3D 0000 cmp ax, 0

78 0027 7E 0C jle mErrorCounter

79 0029 A3 0002r mov counter, ax

80 002C BA 00D4r lea dx, CrLf

81 002F E8 00C0 call outString

82 0032 EB 09 90 jmp mEndLoop

83

84 0035 mErrorCounter:

85 0035 BA 0094r lea dx, msgError

86 0038 E8 00B7 call outString

87 003B EB C8 jmp mInCounter

88

89 003D mEndLoop:

90

91 ;введення елементів масиву

92 003D BA 0071r lea dx, msgArray

93 0040 E8 00AF call outString

94 0043 8B 0E 0002r mov cx, counter

95 0047 33 F6 xor si, si

96

97 0049 mInArray:

98 0049 E8 00AD call outPrompt

99 004C BA 002Fr lea dx, buffer

100 004F B4 0A mov ah, 0ah

101 0051 CD 21 int 21h

102 0053 80 3E 0030r 00 cmp buffer + 1, 0

103 0058 74 EF jz mInArray

104 005A E8 00B2 call convertStrToInt

105 005D 73 08 jnc mElementOk

106 005F BA 0094r lea dx, msgError

107 0062 E8 008D call outString

108 0065 EB E2 jmp mInArray

109 0067 mElementOk:

110 0067 8B FE mov di, si

111 0069 03 FF add di, di

112 006B 89 85 0004r mov A[di], ax

113 006F 46 inc si

114 0070 BA 00D4r lea dx, CrLf

Turbo Assembler Version 3.0 12/21/20 17:37:13 Page 3

ak7.asm

115 0073 E8 007C call outString

116 0076 E2 D1 loop mInArray

117

118 0078 33 F6 xor si, si

119 007A 33 DB xor bx, bx

120 007C 8B 0E 0002r mov cx, counter

121

122 0080 mWhile:

123 0080 F7 84 0004r 0001 test A[si], 1

124 0086 75 0D jnz next

125 0088 8B 84 0004r mov ax, A[si]

126 008C F6 3E 002Cr idiv x

127 0090 89 84 0004r mov A[si], ax

128 0094 43 inc bx

129

130 0095 next:

131 0095 46 inc si

132 0096 46 inc si

133 0097 E2 E7 loop mWhile

134 0099 89 1E 002Dr mov quantity, bx

135

136 ;виведення масиву на консоль

137 009D BA 009Fr lea dx, msgPrint

138 00A0 E8 004F call outString

139 00A3 8B 0E 0002r mov cx, counter

140 00A7 33 F6 xor si, si

141

142 00A9 mPrintArray:

143 00A9 E8 004D call outPrompt

144 00AC 8B FE mov di, si

145 00AE 03 FF add di, di

146 00B0 8B 85 0004r mov ax, A[di]

147 00B4 E8 00B6 call printInt

148 00B7 BA 00D4r lea dx, CrLf

149 00BA E8 0035 call outString

150 00BD 46 inc si ;перехід до наступного елементу +

151 масиву

152 00BE E2 E9 loop mPrintArray ;не всі елементи виведені - на +

153 початок циклу

154

155 ;Вивід кількості замін

156 00C0 BA 00BBr lea dx, msgQuantity

157 00C3 E8 002C call outString

158 00C6 8B 1E 002Dr mov bx, quantity

159 00CA 8B C3 mov ax, bx

160 00CC D4 0A aam

161 00CE 05 3030 add ax,3030h

162 00D1 8A D4 mov dl,ah

163 00D3 8A F0 mov dh,al

164 00D5 B4 02 mov ah,02

165 00D7 CD 21 int 21h

166 00D9 8A D6 mov dl,dh

167 00DB CD 21 int 21h

168 00DD BA 00D4r lea dx, CrLf

169 00E0 E8 000F call outString

170

171

Turbo Assembler Version 3.0 12/21/20 17:37:13 Page 4

ak7.asm

172 ;затримка виконання та вихід з програми

173 00E3 BA 00A9r lea dx, msgPause

174 00E6 E8 0009 call outString

175 00E9 B4 08 mov ah, 08h

176 00EB CD 21 int 21h

177 00ED B8 4C00 mov ax, 4C00h

178 00F0 CD 21 int 21h

179 ;-------------------------------------------------------------------------------

180 ;процедура виведення рядка символів, обмежених +

181 символом '$' на консоль

182 ;вхід: DX - адреса рядка

183 ;-------------------------------------------------------------------------------

184 00F2 outString proc near

185 00F2 50 push ax

186 00F3 B4 09 mov ah, 09h

187 00F5 CD 21 int 21h

188 00F7 58 pop ax

189 00F8 C3 ret

190 00F9 outString endp

191

192 ;-------------------------------------------------------------------------------

193 ;процедура виведення на консоль імені масиву та +

194 індексу елемента

195 ;-------------------------------------------------------------------------------

196 00F9 outPrompt proc near

197 00F9 50 push ax

198 00FA 52 push dx

199 00FB BA 0089r lea dx, msgPrompt1

200 00FE E8 FFF1 call outString

201 0101 8B C6 mov ax, si

202 0103 E8 0067 call printInt

203 0106 BA 008Er lea dx, msgPrompt2

204 0109 E8 FFE6 call outString

205 010C 5A pop dx

206 010D 58 pop ax

207 010E C3 ret

208 010F outPrompt endp

209

210 ;-------------------------------------------------------------------------------

211 ;процедура перетворення рядка в число зі знаком в +

212 форматі слова

213 ;(діапазон значення числа від -32768 до 32767)

214 ;вхід: DX - адреса буфера

215 ;вихід: AX - число в форматі слова (в разі помилки AX = 0)

216 ; CF = 1 - помилка

217 ;-------------------------------------------------------------------------------

218 010F convertStrToInt proc near

219 ;збереження регістрів

220 010F 53 push bx

221 0110 51 push cx

222 0111 52 push dx

223 0112 56 push si

224 0113 57 push di

225 0114 55 push bp

226 0115 8B F2 mov si, dx

227 0117 33 C9 xor cx, cx

228 0119 8A 4C 01 mov cl, [si + 1]

Turbo Assembler Version 3.0 12/21/20 17:37:13 Page 5

ak7.asm

229

230 ;перевірка знаку числа

231 011C 83 C6 02 add si, 2

232 011F 8A 1C mov bl, [si]

233 0121 80 FB 2D cmp bl, '-'

234 0124 75 03 jne mNoSign

235 0126 46 inc si

236 0127 FE C9 dec cl

237

238 ;перетворення рядка у слово без знаку

239 0129 mNoSign:

240 0129 8B EB mov bp, bx

241 012B 33 DB xor bx, bx

242 012D BF 000A mov di, 10

243 0130 33 C0 xor ax, ax

244 0132 mScanString:

245 0132 8A 1C mov bl, [si]

246 0134 46 inc si

247 0135 80 FB 30 cmp bl, '0'

248 0138 72 29 jb mErrorValue

249 013A 80 FB 39 cmp bl, '9'

250 013D 77 24 ja mErrorValue

251 013F 80 EB 30 sub bl, '0'

252 0142 F7 E7 mul di ;AX

253 0144 72 1D jc mErrorValue

254 0146 03 C3 add ax, bx

255 0148 72 19 jc mErrorValue

256 014A E2 E6 loop mScanString

257

258 ;обробка знаку числа та перевірка діапазону для +

259 від'ємного числа

260 014C 8B DD mov bx, bp

261 014E 80 FB 2D cmp bl, '-'

262 0151 75 07 jne mPlus

263 0153 3D 8000 cmp ax, 32768

264 0156 77 0B ja mErrorValue

265 0158 F7 D8 neg ax

266

267 ;перевірка діапазону для додатного числа

268 015A mPlus:

269 015A 3D 7FFF cmp ax, 32767

270 015D 77 04 ja mErrorValue

271

272 ;успішне перетворення рядка у число в форматі слова

273 015F mOk:

274 015F F8 clc

275 0160 EB 04 90 jmp mExit

276

277 ;помилка перетворення рядка у число в форматі слова

278 0163 mErrorValue:

279 0163 33 C0 xor ax, ax

280 0165 F9 stc

281

282 0166 mExit:

283 ;відновлення регістрів

284 0166 5D pop bp

285 0167 5F pop di

Turbo Assembler Version 3.0 12/21/20 17:37:13 Page 6

ak7.asm

286 0168 5E pop si

287 0169 5A pop dx

288 016A 59 pop cx

289 016B 5B pop bx

290 016C C3 ret

291 016D convertStrToInt endp

292

293 ;-------------------------------------------------------------------------------

294 ;процедура виведення цілого числа у форматі слова +

295 на консоль

296 ;вхід: AX - ціле число

297 ;-------------------------------------------------------------------------------

298 016D printInt proc near

299 ;збереження регістрів

300 016D 50 push ax

301 016E 53 push bx

302 016F 51 push cx

303 0170 52 push dx

304

305 ;обробка знаку числа AX

306 0171 3D 0000 cmp ax, 0

307 0174 7D 0A jge mPositive

308 0176 F7 D8 neg ax

309 0178 50 push ax

310 0179 B2 2D mov dl, '-'

311 017B B4 02 mov ah, 02h

312 017D CD 21 int 21h

313 017F 58 pop ax

314

315 ;обробка додатного числа

316 0180 mPositive:

317 0180 BB 000A mov bx, 10

318 0183 33 C9 xor cx, cx

319

320 ;отримуємо цифри числа AX

321 0185 mDivisionNumber:

322 0185 99 cwd

323 0186 F7 F3 div bx

324 0188 52 push dx

325 0189 41 inc cx

326 018A 0B C0 or ax, ax

327 018C 75 F7 jnz mDivisionNumber

328

329 ;виведення на консоль цифр числа AX

330 018E mPrintChar:

331 018E 5A pop dx

332 018F 80 C2 30 add dl, 30h

333 0192 B4 02 mov ah, 02h

334 0194 CD 21 int 21h

335 0196 E2 F6 loop mPrintChar

336

337 ;відновлення регістрів

338 0198 5A pop dx

339 0199 59 pop cx

340 019A 5B pop bx

341 019B 58 pop ax

342 019C C3 ret

Turbo Assembler Version 3.0 12/21/20 17:37:13 Page 7

ak7.asm

343 019D printInt endp

344 ;завершення програми

345 019D B8 4C00 mov ax, 4C00h

346 01A0 CD 21 int 21h

347 01A2 code ends

348 end begin

Turbo Assembler Version 3.0 12/21/20 17:37:13 Page 8

Symbol Table

Symbol Name Type Value

??DATE Text "12/21/20"

??FILENAME Text "ak7 "

??TIME Text "17:37:13"

??VERSION Number 0300

@CPU Text 0101H

@CURSEG Text CODE

@FILENAME Text AK7

@WORDSIZE Text 2

A Word DATA:0004

BEGIN Near CODE:0000

BUFFER Byte DATA:002F

CONVERTSTRTOINT Near CODE:010F

COUNTER Word DATA:0002

CRLF Byte DATA:00D4

LEN Word DATA:0000

MDIVISIONNUMBER Near CODE:0185

MELEMENTOK Near CODE:0067

MENDLOOP Near CODE:003D

MERRORCOUNTER Near CODE:0035

MERRORVALUE Near CODE:0163

MEXIT Near CODE:0166

MINARRAY Near CODE:0049

MINCOUNTER Near CODE:0005

MNOSIGN Near CODE:0129

MOK Near CODE:015F

MPLUS Near CODE:015A

MPOSITIVE Near CODE:0180

MPRINTARRAY Near CODE:00A9

MPRINTCHAR Near CODE:018E

MSCANSTRING Near CODE:0132

MSGARRAY Byte DATA:0071

MSGCOUNTER Byte DATA:0038

MSGERROR Byte DATA:0094

MSGPAUSE Byte DATA:00A9

MSGPRINT Byte DATA:009F

MSGPROMPT1 Byte DATA:0089

MSGPROMPT2 Byte DATA:008E

MSGQUANTITY Byte DATA:00BB

MWHILE Near CODE:0080

NEXT Near CODE:0095

OUTPROMPT Near CODE:00F9

OUTSTRING Near CODE:00F2

PRINTINT Near CODE:016D

QUANTITY Word DATA:002D

X Byte DATA:002C

Groups & Segments Bit Size Align Combine Class

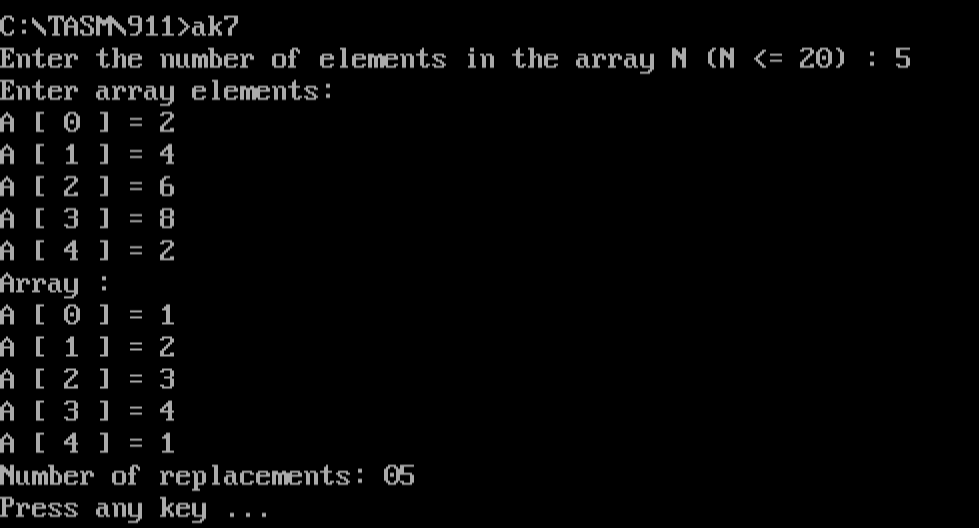
CODE 16 01A2 Para Public CODE

DATA 16 00D7 Para Public DATA

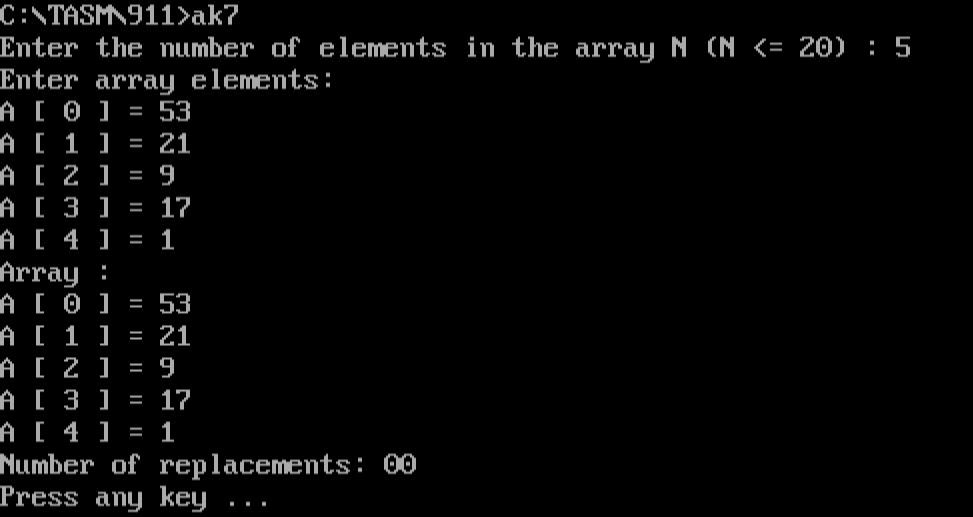
STK 16 0080 Para Stack

**5. Результати тестувань**

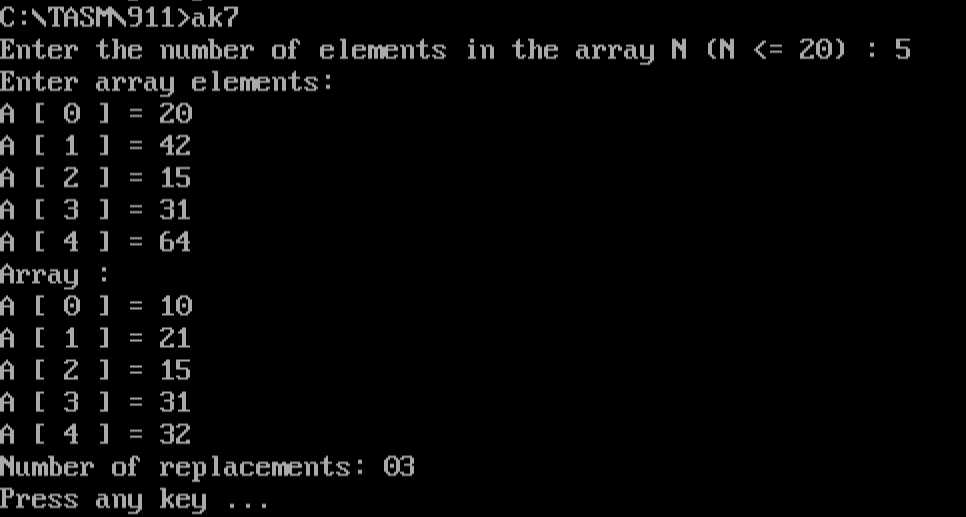
5. 1. Усі числа парні



5. 2. Усі числа непарні



5. 3. Парні та непарні числа

****

Як видно на зображення всі результати тестів збіглися з очікуваними, програма працює вірно.

**6. Висновок**

Під час лабараторної роботи працював з масивом чисел. Для введеня в ассемблері використовуються ASCII коди, для того, щоб цифру із символьного виду перетворити в число, необхідно додати до ASCII коду 30h, і відняти 30h, щоб зробити це у зворотньому напрямі. Вивід двовимірного масива на екран відбувається за рахунок програміста, адже в асемблері немає спеціальних засобів розпізнавання двовимірного масиву. Для доступу до елементів масиву використовують базово-індексну адресацію, та базово-індексну адресацію з сувом.