НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування та спеціалізованих комп’ютерних систем

Лабораторна робота №2

з дисципліни

«Системне програмування»

Тема: «Реалізація основних програмних конструктів мовою Асемблера.

Використання асемблерних вставок у програмах мовою С++»

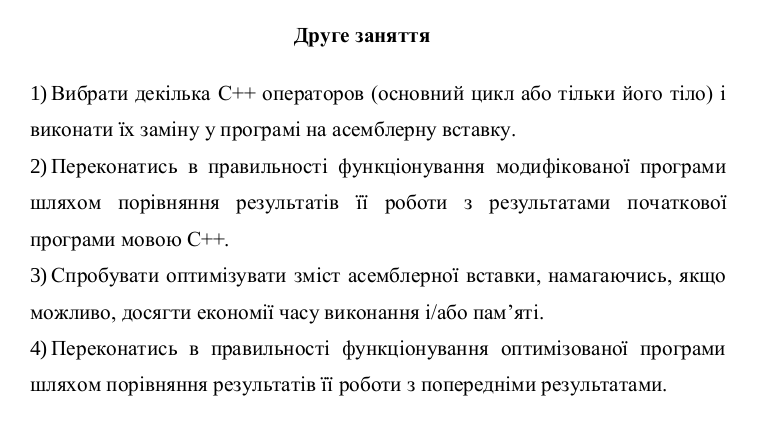
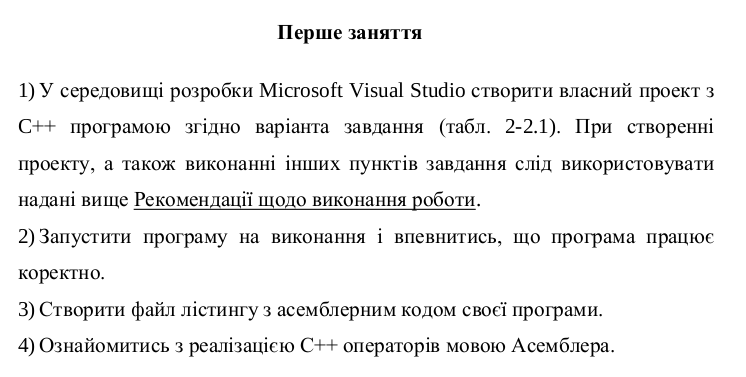
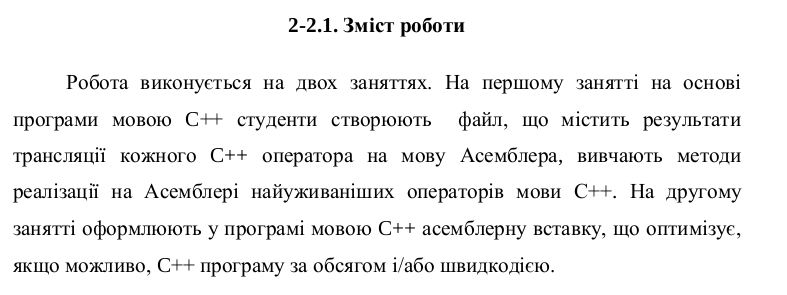
Виконав студент 2 курсу

ФПМ групи КВ-71

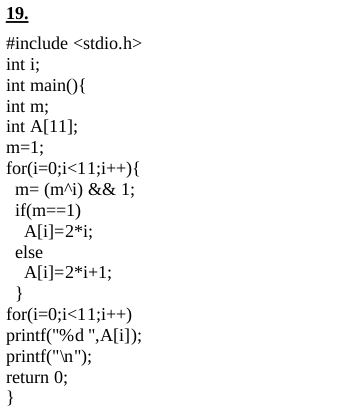
Рибак Ю. О.

Перевірив:

Київ – 2018



**Варіант** : 19



**Текст програми:**

**С++:**

#include "pch.h"

#include <stdio.h>

int i;

int main() {

int m = 1;

int A[11];

for (i = 0; i < 11; i++) {

m = (m^i) && 1;

if (m == 1) A[i] = 2 \* i;

else A[i] = 2 \* i + 1;

}

for (i = 0; i < 11; i++) printf("%d ", A[i]);

printf("\n");

return 0;

}

**Вставка:**

#include <stdio.h>

#include <iostream>

int i;

int main() {

int m = 1;

int A[11];

int tv69 = 0;

for (i = 0; i < 11; i++) {

\_asm {

mov m, 1

mov i, 0

jmp LN4

LN2 :

mov eax, i

add eax, 1

mov i, eax

LN4 :

cmp i, 11

jge LN3

mov eax, m

xor eax, i

je LN8

mov ecx, 1

test ecx, ecx

je LN8

mov tv69, 1

jmp LN9

LN8 :

mov tv69, 0

LN9 :

mov edx, tv69

mov m, edx

cmp m, 1

jne LN5

mov eax, i

shl eax, 1

mov ecx, i

mov A[ecx \* 4], eax

jmp LN6

LN5 :

mov eax, i

lea ecx, DWORD PTR[eax + eax + 1]

mov edx, i

mov A[edx \* 4], ecx

LN6 :

jmp LN2

LN3 :

xor eax, eax

}

}

for (i = 0; i < 11; i++) printf("%d ", A[i]);

printf("\n");

getc(stdin);

return 0;

}

**.code:**

; Listing generated by Microsoft (R) Optimizing Compiler Version 19.15.26730.0

TITLE C:\Users\yurar\Desktop\Нова папка\task1\ConsoleApplication1\ConsoleApplication1\ConsoleApplication1.c

.686P

.XMM

include listing.inc

.model flat

INCLUDELIB MSVCRTD

INCLUDELIB OLDNAMES

\_DATA SEGMENT

COMM \_i:DWORD

\_DATA ENDS

msvcjmc SEGMENT

\_\_5D0131B6\_pch@h DB 01H

\_\_8862E326\_consoleapplication1@pch DB 01H

\_\_02F148FA\_consoleapplication1@c DB 01H

\_\_A3797CDC\_stdio@h DB 01H

\_\_BAC7FC50\_corecrt\_wstdio@h DB 01H

\_\_320E01E0\_corecrt\_stdio\_config@h DB 01H

msvcjmc ENDS

PUBLIC \_\_\_local\_stdio\_printf\_options

PUBLIC \_\_vfprintf\_l

PUBLIC \_printf

PUBLIC \_main

PUBLIC \_\_JustMyCode\_Default

PUBLIC ??\_C@\_03JDANDILB@?$CFd?5@ ; `string'

PUBLIC ??\_C@\_01EEMJAFIK@?6@ ; `string'

EXTRN \_\_imp\_\_\_\_acrt\_iob\_func:PROC

EXTRN \_\_imp\_\_\_\_stdio\_common\_vfprintf:PROC

EXTRN @\_RTC\_CheckStackVars@8:PROC

EXTRN @\_\_CheckForDebuggerJustMyCode@4:PROC

EXTRN \_\_RTC\_CheckEsp:PROC

EXTRN \_\_RTC\_InitBase:PROC

EXTRN \_\_RTC\_Shutdown:PROC

\_DATA SEGMENT

COMM ?\_OptionsStorage@?1??\_\_local\_stdio\_printf\_options@@9@9:QWORD ; `\_\_local\_stdio\_printf\_options'::`2'::\_OptionsStorage

\_DATA ENDS

; COMDAT rtc$TMZ

rtc$TMZ SEGMENT

\_\_RTC\_Shutdown.rtc$TMZ DD FLAT:\_\_RTC\_Shutdown

rtc$TMZ ENDS

; COMDAT rtc$IMZ

rtc$IMZ SEGMENT

\_\_RTC\_InitBase.rtc$IMZ DD FLAT:\_\_RTC\_InitBase

rtc$IMZ ENDS

; COMDAT ??\_C@\_01EEMJAFIK@?6@

CONST SEGMENT

??\_C@\_01EEMJAFIK@?6@ DB 0aH, 00H ; `string'

CONST ENDS

; COMDAT ??\_C@\_03JDANDILB@?$CFd?5@

CONST SEGMENT

??\_C@\_03JDANDILB@?$CFd?5@ DB '%d ', 00H ; `string'

CONST ENDS

; Function compile flags: /Odt

; COMDAT \_\_JustMyCode\_Default

\_TEXT SEGMENT

\_\_JustMyCode\_Default PROC ; COMDAT

00000 55 push ebp

00001 8b ec mov ebp, esp

00003 5d pop ebp

00004 c3 ret 0

\_\_JustMyCode\_Default ENDP

\_TEXT ENDS

; Function compile flags: /Odtp /RTCsu /ZI

; File c:\users\yurar\desktop\нова папка\task1\consoleapplication1\consoleapplication1\consoleapplication1.c

; COMDAT \_main

\_TEXT SEGMENT

tv69 = -260 ; size = 4

\_A$ = -60 ; size = 44

\_m$ = -8 ; size = 4

\_main PROC ; COMDAT

00000 55 push ebp

00001 8b ec mov ebp, esp

00003 81 ec 04 01 00

00 sub esp, 260 ; 00000104H

00009 53 push ebx

0000a 56 push esi

0000b 57 push edi

0000c 8d bd fc fe ff

ff lea edi, DWORD PTR [ebp-260]

00012 b9 41 00 00 00 mov ecx, 65 ; 00000041H

00017 b8 cc cc cc cc mov eax, -858993460 ; ccccccccH

0001c f3 ab rep stosd

0001e b9 00 00 00 00 mov ecx, OFFSET \_\_02F148FA\_consoleapplication1@c

00023 e8 00 00 00 00 call @\_\_CheckForDebuggerJustMyCode@4

00028 c7 45 f8 01 00

00 00 mov DWORD PTR \_m$[ebp], 1

0002f c7 05 00 00 00

00 00 00 00 00 mov DWORD PTR \_i, 0

00039 eb 0d jmp SHORT $LN4@main

$LN2@main:

0003b a1 00 00 00 00 mov eax, DWORD PTR \_i

00040 83 c0 01 add eax, 1

00043 a3 00 00 00 00 mov DWORD PTR \_i, eax

$LN4@main:

00048 83 3d 00 00 00

00 0b cmp DWORD PTR \_i, 11 ; 0000000bH

0004f 7d 61 jge SHORT $LN3@main

00051 8b 45 f8 mov eax, DWORD PTR \_m$[ebp]

00054 33 05 00 00 00

00 xor eax, DWORD PTR \_i

0005a 74 15 je SHORT $LN11@main

0005c b9 01 00 00 00 mov ecx, 1

00061 85 c9 test ecx, ecx

00063 74 0c je SHORT $LN11@main

00065 c7 85 fc fe ff

ff 01 00 00 00 mov DWORD PTR tv69[ebp], 1

0006f eb 0a jmp SHORT $LN12@main

$LN11@main:

00071 c7 85 fc fe ff

ff 00 00 00 00 mov DWORD PTR tv69[ebp], 0

$LN12@main:

0007b 8b 95 fc fe ff

ff mov edx, DWORD PTR tv69[ebp]

00081 89 55 f8 mov DWORD PTR \_m$[ebp], edx

00084 83 7d f8 01 cmp DWORD PTR \_m$[ebp], 1

00088 75 13 jne SHORT $LN8@main

0008a a1 00 00 00 00 mov eax, DWORD PTR \_i

0008f d1 e0 shl eax, 1

00091 8b 0d 00 00 00

00 mov ecx, DWORD PTR \_i

00097 89 44 8d c4 mov DWORD PTR \_A$[ebp+ecx\*4], eax

0009b eb 13 jmp SHORT $LN9@main

$LN8@main:

0009d a1 00 00 00 00 mov eax, DWORD PTR \_i

000a2 8d 4c 00 01 lea ecx, DWORD PTR [eax+eax+1]

000a6 8b 15 00 00 00

00 mov edx, DWORD PTR \_i

000ac 89 4c 95 c4 mov DWORD PTR \_A$[ebp+edx\*4], ecx

$LN9@main:

000b0 eb 89 jmp SHORT $LN2@main

$LN3@main:

000b2 c7 05 00 00 00

00 00 00 00 00 mov DWORD PTR \_i, 0

000bc eb 0d jmp SHORT $LN7@main

$LN5@main:

000be a1 00 00 00 00 mov eax, DWORD PTR \_i

000c3 83 c0 01 add eax, 1

000c6 a3 00 00 00 00 mov DWORD PTR \_i, eax

$LN7@main:

000cb 83 3d 00 00 00

00 0b cmp DWORD PTR \_i, 11 ; 0000000bH

000d2 7d 19 jge SHORT $LN6@main

000d4 a1 00 00 00 00 mov eax, DWORD PTR \_i

000d9 8b 4c 85 c4 mov ecx, DWORD PTR \_A$[ebp+eax\*4]

000dd 51 push ecx

000de 68 00 00 00 00 push OFFSET ??\_C@\_03JDANDILB@?$CFd?5@

000e3 e8 00 00 00 00 call \_printf

000e8 83 c4 08 add esp, 8

000eb eb d1 jmp SHORT $LN5@main

$LN6@main:

000ed 68 00 00 00 00 push OFFSET ??\_C@\_01EEMJAFIK@?6@

000f2 e8 00 00 00 00 call \_printf

000f7 83 c4 04 add esp, 4

000fa 33 c0 xor eax, eax

000fc 52 push edx

000fd 8b cd mov ecx, ebp

000ff 50 push eax

00100 8d 15 00 00 00

00 lea edx, DWORD PTR $LN15@main

00106 e8 00 00 00 00 call @\_RTC\_CheckStackVars@8

0010b 58 pop eax

0010c 5a pop edx

0010d 5f pop edi

0010e 5e pop esi

0010f 5b pop ebx

00110 81 c4 04 01 00

00 add esp, 260 ; 00000104H

00116 3b ec cmp ebp, esp

00118 e8 00 00 00 00 call \_\_RTC\_CheckEsp

0011d 8b e5 mov esp, ebp

0011f 5d pop ebp

00120 c3 ret 0

00121 0f 1f 00 npad 3

$LN15@main:

00124 01 00 00 00 DD 1

00128 00 00 00 00 DD $LN14@main

$LN14@main:

0012c c4 ff ff ff DD -60 ; ffffffc4H

00130 2c 00 00 00 DD 44 ; 0000002cH

00134 00 00 00 00 DD $LN13@main

$LN13@main:

00138 41 DB 65 ; 00000041H

00139 00 DB 0

\_main ENDP

\_TEXT ENDS

; Function compile flags: /Odtp /RTCsu /ZI

; File c:\program files (x86)\windows kits\10\include\10.0.17134.0\ucrt\stdio.h

; COMDAT \_printf

\_TEXT SEGMENT

\_\_ArgList$ = -20 ; size = 4

\_\_Result$ = -8 ; size = 4

\_\_Format$ = 8 ; size = 4

\_printf PROC ; COMDAT

; 954 : {

00000 55 push ebp

00001 8b ec mov ebp, esp

00003 81 ec d8 00 00

00 sub esp, 216 ; 000000d8H

00009 53 push ebx

0000a 56 push esi

0000b 57 push edi

0000c 8d bd 28 ff ff

ff lea edi, DWORD PTR [ebp-216]

00012 b9 36 00 00 00 mov ecx, 54 ; 00000036H

00017 b8 cc cc cc cc mov eax, -858993460 ; ccccccccH

0001c f3 ab rep stosd

0001e b9 00 00 00 00 mov ecx, OFFSET \_\_A3797CDC\_stdio@h

00023 e8 00 00 00 00 call @\_\_CheckForDebuggerJustMyCode@4

; 955 : int \_Result;

; 956 : va\_list \_ArgList;

; 957 : \_\_crt\_va\_start(\_ArgList, \_Format);

00028 8d 45 0c lea eax, DWORD PTR \_\_Format$[ebp+4]

0002b 89 45 ec mov DWORD PTR \_\_ArgList$[ebp], eax

; 958 : \_Result = \_vfprintf\_l(stdout, \_Format, NULL, \_ArgList);

0002e 8b 45 ec mov eax, DWORD PTR \_\_ArgList$[ebp]

00031 50 push eax

00032 6a 00 push 0

00034 8b 4d 08 mov ecx, DWORD PTR \_\_Format$[ebp]

00037 51 push ecx

00038 8b f4 mov esi, esp

0003a 6a 01 push 1

0003c ff 15 00 00 00

00 call DWORD PTR \_\_imp\_\_\_\_acrt\_iob\_func

00042 83 c4 04 add esp, 4

00045 3b f4 cmp esi, esp

00047 e8 00 00 00 00 call \_\_RTC\_CheckEsp

0004c 50 push eax

0004d e8 00 00 00 00 call \_\_vfprintf\_l

00052 83 c4 10 add esp, 16 ; 00000010H

00055 89 45 f8 mov DWORD PTR \_\_Result$[ebp], eax

; 959 : \_\_crt\_va\_end(\_ArgList);

00058 c7 45 ec 00 00

00 00 mov DWORD PTR \_\_ArgList$[ebp], 0

; 960 : return \_Result;

0005f 8b 45 f8 mov eax, DWORD PTR \_\_Result$[ebp]

; 961 : }

00062 5f pop edi

00063 5e pop esi

00064 5b pop ebx

00065 81 c4 d8 00 00

00 add esp, 216 ; 000000d8H

0006b 3b ec cmp ebp, esp

0006d e8 00 00 00 00 call \_\_RTC\_CheckEsp

00072 8b e5 mov esp, ebp

00074 5d pop ebp

00075 c3 ret 0

\_printf ENDP

\_TEXT ENDS

; Function compile flags: /Odtp /RTCsu /ZI

; File c:\program files (x86)\windows kits\10\include\10.0.17134.0\ucrt\stdio.h

; COMDAT \_\_vfprintf\_l

\_TEXT SEGMENT

\_\_Stream$ = 8 ; size = 4

\_\_Format$ = 12 ; size = 4

\_\_Locale$ = 16 ; size = 4

\_\_ArgList$ = 20 ; size = 4

\_\_vfprintf\_l PROC ; COMDAT

; 642 : {

00000 55 push ebp

00001 8b ec mov ebp, esp

00003 81 ec c0 00 00

00 sub esp, 192 ; 000000c0H

00009 53 push ebx

0000a 56 push esi

0000b 57 push edi

0000c 8d bd 40 ff ff

ff lea edi, DWORD PTR [ebp-192]

00012 b9 30 00 00 00 mov ecx, 48 ; 00000030H

00017 b8 cc cc cc cc mov eax, -858993460 ; ccccccccH

0001c f3 ab rep stosd

0001e b9 00 00 00 00 mov ecx, OFFSET \_\_A3797CDC\_stdio@h

00023 e8 00 00 00 00 call @\_\_CheckForDebuggerJustMyCode@4

; 643 : return \_\_stdio\_common\_vfprintf(\_CRT\_INTERNAL\_LOCAL\_PRINTF\_OPTIONS, \_Stream, \_Format, \_Locale, \_ArgList);

00028 8b f4 mov esi, esp

0002a 8b 45 14 mov eax, DWORD PTR \_\_ArgList$[ebp]

0002d 50 push eax

0002e 8b 4d 10 mov ecx, DWORD PTR \_\_Locale$[ebp]

00031 51 push ecx

00032 8b 55 0c mov edx, DWORD PTR \_\_Format$[ebp]

00035 52 push edx

00036 8b 45 08 mov eax, DWORD PTR \_\_Stream$[ebp]

00039 50 push eax

0003a e8 00 00 00 00 call \_\_\_local\_stdio\_printf\_options

0003f 8b 48 04 mov ecx, DWORD PTR [eax+4]

00042 51 push ecx

00043 8b 10 mov edx, DWORD PTR [eax]

00045 52 push edx

00046 ff 15 00 00 00

00 call DWORD PTR \_\_imp\_\_\_\_stdio\_common\_vfprintf

0004c 83 c4 18 add esp, 24 ; 00000018H

0004f 3b f4 cmp esi, esp

00051 e8 00 00 00 00 call \_\_RTC\_CheckEsp

; 644 : }

00056 5f pop edi

00057 5e pop esi

00058 5b pop ebx

00059 81 c4 c0 00 00

00 add esp, 192 ; 000000c0H

0005f 3b ec cmp ebp, esp

00061 e8 00 00 00 00 call \_\_RTC\_CheckEsp

00066 8b e5 mov esp, ebp

00068 5d pop ebp

00069 c3 ret 0

\_\_vfprintf\_l ENDP

\_TEXT ENDS

; Function compile flags: /Odtp /RTCsu /ZI

; File c:\program files (x86)\windows kits\10\include\10.0.17134.0\ucrt\corecrt\_stdio\_config.h

; COMDAT \_\_\_local\_stdio\_printf\_options

\_TEXT SEGMENT

\_\_\_local\_stdio\_printf\_options PROC ; COMDAT

; 85 : {

00000 55 push ebp

00001 8b ec mov ebp, esp

00003 81 ec c0 00 00

00 sub esp, 192 ; 000000c0H

00009 53 push ebx

0000a 56 push esi

0000b 57 push edi

0000c 8d bd 40 ff ff

ff lea edi, DWORD PTR [ebp-192]

00012 b9 30 00 00 00 mov ecx, 48 ; 00000030H

00017 b8 cc cc cc cc mov eax, -858993460 ; ccccccccH

0001c f3 ab rep stosd

0001e b9 00 00 00 00 mov ecx, OFFSET \_\_320E01E0\_corecrt\_stdio\_config@h

00023 e8 00 00 00 00 call @\_\_CheckForDebuggerJustMyCode@4

; 86 : static unsigned \_\_int64 \_OptionsStorage;

; 87 : return &\_OptionsStorage;

00028 b8 00 00 00 00 mov eax, OFFSET ?\_OptionsStorage@?1??\_\_local\_stdio\_printf\_options@@9@9 ; `\_\_local\_stdio\_printf\_options'::`2'::\_OptionsStorage

; 88 : }

0002d 5f pop edi

0002e 5e pop esi

0002f 5b pop ebx

00030 81 c4 c0 00 00

00 add esp, 192 ; 000000c0H

00036 3b ec cmp ebp, esp

00038 e8 00 00 00 00 call \_\_RTC\_CheckEsp

0003d 8b e5 mov esp, ebp

0003f 5d pop ebp

00040 c3 ret 0

\_\_\_local\_stdio\_printf\_options ENDP

\_TEXT ENDS

END