****

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

**Факультет прикладної математики**

**Кафедра системного програмування і 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++»**

**Варіант 5**

**Виконав:**

**студент II-го курсу**

**групи КВ-41**

**Горпинич-Радуженко Іван**

**Київ 2015**

**Завдання на виконання роботи**

1) У середовищі розробки Microsoft Visual Studio створити власний проект з C++ програмою згідно варіанта завдання . При створенні

проекту, а також виконанні інших пунктів завдання слід використовувати

надані вище Рекомендації щодо виконання роботи.

2) Запустити програму на виконання і впевнитись, що програма працює

коректно.

3) Створити файл лістингу з асемблерним кодом своєї програми.

4) Ознайомитись з реалізацією С++ операторів мовою Асемблера.

5) Вибрати декілька С++ операторов (основний цикл або тільки його тіло) і

виконати їх заміну у програмі на асемблерну вставку.

6) Переконатись в правильності функціонування модифікованої програми

шляхом порівняння результатів її роботи з результатами початкової

програми мовою С++.

7) Спробувати оптимізувати зміст асемблерної вставки, намагаючись, якщо можливо, досягти економії часу виконання і/або пам‘яті.

8) Переконатись в правильності функціонування оптимізованої програми шляхом порівняння результатів її роботи з попередніми результатами.

**Варіант 5**

**С++ програма:**

#include <stdio.h>

int A[7];

int d;

int main() {

int i;

d = 1;

i = 0;

while (i <= 6) {

A[i] = i | d;

d += A[i];

if (d>10)

A[i] = 127 - i;

i++;

}

for (i = 0;i<7;i++)

printf("%d ", A[i]);

printf("\n");

return 0;

**Лістинг №1**

; Listing generated by Microsoft (R) Optimizing Compiler Version 19.00.23026.0

TITLE D:\Projects\Lab2-5\Lab2-5\Lab2-5.cpp

.686P

.XMM

include listing.inc

.model flat

INCLUDELIB MSVCRTD

INCLUDELIB OLDNAMES

PUBLIC ?A@@3PAHA ; A

PUBLIC ?d@@3HA ; d

\_BSS SEGMENT

?A@@3PAHA DD 07H DUP (?) ; A

?d@@3HA DD 01H DUP (?) ; d

\_BSS ENDS

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

PUBLIC \_\_vfprintf\_l

PUBLIC ??$\_\_vcrt\_va\_start\_verify\_argument\_type@QBD@@YAXXZ ; \_\_vcrt\_va\_start\_verify\_argument\_type<char const \* const>

PUBLIC \_printf

PUBLIC \_main

PUBLIC ?\_OptionsStorage@?1??\_\_local\_stdio\_printf\_options@@9@4\_KA ; `\_\_local\_stdio\_printf\_options'::`2'::\_OptionsStorage

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

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

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

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

EXTRN \_\_RTC\_CheckEsp:PROC

EXTRN \_\_RTC\_InitBase:PROC

EXTRN \_\_RTC\_Shutdown:PROC

; COMDAT ?\_OptionsStorage@?1??\_\_local\_stdio\_printf\_options@@9@4\_KA

\_BSS SEGMENT

?\_OptionsStorage@?1??\_\_local\_stdio\_printf\_options@@9@4\_KA DQ 01H DUP (?) ; `\_\_local\_stdio\_printf\_options'::`2'::\_OptionsStorage

\_BSS 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?$AA@

CONST SEGMENT

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

CONST ENDS

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

CONST SEGMENT

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

CONST ENDS

; Function compile flags: /Odtp /RTCsu /ZI

; File d:\projects\lab2-5\lab2-5\lab2-5.cpp

; COMDAT \_main

\_TEXT SEGMENT

\_i$ = -8 ; size = 4

\_main PROC ; COMDAT

; 4 : int main() {

00000 55 push ebp

00001 8b ec mov ebp, esp

00003 81 ec cc 00 00

00 sub esp, 204 ; 000000ccH

00009 53 push ebx

0000a 56 push esi

0000b 57 push edi

0000c 8d bd 34 ff ff

ff lea edi, DWORD PTR [ebp-204]

00012 b9 33 00 00 00 mov ecx, 51 ; 00000033H

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

0001c f3 ab rep stosd

; 5 : int i;

; 6 : d = 1;

0001e c7 05 00 00 00

00 01 00 00 00 mov DWORD PTR ?d@@3HA, 1 ; d

; 7 : i = 0;

00028 c7 45 f8 00 00

00 00 mov DWORD PTR \_i$[ebp], 0

$LN2@main:

; 8 : while (i <= 6) {

0002f 83 7d f8 06 cmp DWORD PTR \_i$[ebp], 6

00033 7f 4f jg SHORT $LN3@main

; 9 : A[i] = i | d;

00035 8b 45 f8 mov eax, DWORD PTR \_i$[ebp]

00038 0b 05 00 00 00

00 or eax, DWORD PTR ?d@@3HA ; d

0003e 8b 4d f8 mov ecx, DWORD PTR \_i$[ebp]

00041 89 04 8d 00 00

00 00 mov DWORD PTR ?A@@3PAHA[ecx\*4], eax

; 10 : d += A[i];

00048 8b 45 f8 mov eax, DWORD PTR \_i$[ebp]

0004b 8b 0d 00 00 00

00 mov ecx, DWORD PTR ?d@@3HA ; d

00051 03 0c 85 00 00

00 00 add ecx, DWORD PTR ?A@@3PAHA[eax\*4]

00058 89 0d 00 00 00

00 mov DWORD PTR ?d@@3HA, ecx ; d

; 11 : if (d>10)

0005e 83 3d 00 00 00

00 0a cmp DWORD PTR ?d@@3HA, 10 ; d, 0000000aH

00065 7e 12 jle SHORT $LN7@main

; 12 : A[i] = 127 - i;

00067 b8 7f 00 00 00 mov eax, 127 ; 0000007fH

0006c 2b 45 f8 sub eax, DWORD PTR \_i$[ebp]

0006f 8b 4d f8 mov ecx, DWORD PTR \_i$[ebp]

00072 89 04 8d 00 00

00 00 mov DWORD PTR ?A@@3PAHA[ecx\*4], eax

$LN7@main:

; 13 : i++;

00079 8b 45 f8 mov eax, DWORD PTR \_i$[ebp]

0007c 83 c0 01 add eax, 1

0007f 89 45 f8 mov DWORD PTR \_i$[ebp], eax

; 14 : }

00082 eb ab jmp SHORT $LN2@main

$LN3@main:

; 15 : for (i = 0;i<7;i++)

00084 c7 45 f8 00 00

00 00 mov DWORD PTR \_i$[ebp], 0

0008b eb 09 jmp SHORT $LN6@main

$LN4@main:

0008d 8b 45 f8 mov eax, DWORD PTR \_i$[ebp]

00090 83 c0 01 add eax, 1

00093 89 45 f8 mov DWORD PTR \_i$[ebp], eax

$LN6@main:

00096 83 7d f8 07 cmp DWORD PTR \_i$[ebp], 7

0009a 7d 1a jge SHORT $LN5@main

; 16 : printf("%d ", A[i]);

0009c 8b 45 f8 mov eax, DWORD PTR \_i$[ebp]

0009f 8b 0c 85 00 00

00 00 mov ecx, DWORD PTR ?A@@3PAHA[eax\*4]

000a6 51 push ecx

000a7 68 00 00 00 00 push OFFSET ??\_C@\_03JDANDILB@?$CFd?5?$AA@

000ac e8 00 00 00 00 call \_printf

000b1 83 c4 08 add esp, 8

000b4 eb d7 jmp SHORT $LN4@main

$LN5@main:

; 17 : printf("\n");

000b6 68 00 00 00 00 push OFFSET ??\_C@\_01EEMJAFIK@?6?$AA@

000bb e8 00 00 00 00 call \_printf

000c0 83 c4 04 add esp, 4

; 18 : return 0;

000c3 33 c0 xor eax, eax

; 19 : }

000c5 5f pop edi

000c6 5e pop esi

000c7 5b pop ebx

000c8 81 c4 cc 00 00

00 add esp, 204 ; 000000ccH

000ce 3b ec cmp ebp, esp

000d0 e8 00 00 00 00 call \_\_RTC\_CheckEsp

000d5 8b e5 mov esp, ebp

000d7 5d pop ebp

000d8 c3 ret 0

\_main ENDP

\_TEXT ENDS

; Function compile flags: /Odtp /RTCsu /ZI

; File c:\program files (x86)\windows kits\10\include\10.0.10150.0\ucrt\stdio.h

; COMDAT \_printf

\_TEXT SEGMENT

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

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

\_\_Format$ = 8 ; size = 4

\_printf PROC ; COMDAT

; 950 : {

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

; 951 : int \_Result;

; 952 : va\_list \_ArgList;

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

0001e e8 00 00 00 00 call ??$\_\_vcrt\_va\_start\_verify\_argument\_type@QBD@@YAXXZ ; \_\_vcrt\_va\_start\_verify\_argument\_type<char const \* const>

00023 8d 45 0c lea eax, DWORD PTR \_\_Format$[ebp+4]

00026 89 45 ec mov DWORD PTR \_\_ArgList$[ebp], eax

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

00029 8b 45 ec mov eax, DWORD PTR \_\_ArgList$[ebp]

0002c 50 push eax

0002d 6a 00 push 0

0002f 8b 4d 08 mov ecx, DWORD PTR \_\_Format$[ebp]

00032 51 push ecx

00033 8b f4 mov esi, esp

00035 6a 01 push 1

00037 ff 15 00 00 00

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

0003d 83 c4 04 add esp, 4

00040 3b f4 cmp esi, esp

00042 e8 00 00 00 00 call \_\_RTC\_CheckEsp

00047 50 push eax

00048 e8 00 00 00 00 call \_\_vfprintf\_l

0004d 83 c4 10 add esp, 16 ; 00000010H

00050 89 45 f8 mov DWORD PTR \_\_Result$[ebp], eax

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

00053 c7 45 ec 00 00

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

; 956 : return \_Result;

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

; 957 : }

0005d 5f pop edi

0005e 5e pop esi

0005f 5b pop ebx

00060 81 c4 d8 00 00

00 add esp, 216 ; 000000d8H

00066 3b ec cmp ebp, esp

00068 e8 00 00 00 00 call \_\_RTC\_CheckEsp

0006d 8b e5 mov esp, ebp

0006f 5d pop ebp

00070 c3 ret 0

\_printf ENDP

\_TEXT ENDS

; Function compile flags: /Odtp /RTCsu /ZI

; File d:\program files (x86)\vs\vc\include\vadefs.h

; COMDAT ??$\_\_vcrt\_va\_start\_verify\_argument\_type@QBD@@YAXXZ

\_TEXT SEGMENT

??$\_\_vcrt\_va\_start\_verify\_argument\_type@QBD@@YAXXZ PROC ; \_\_vcrt\_va\_start\_verify\_argument\_type<char const \* const>, COMDAT

; 144 : {

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

; 145 : static\_assert(!\_\_vcrt\_va\_list\_is\_reference<\_Ty>::\_\_the\_value, "va\_start argument must not have reference type and must not be parenthesized");

; 146 : }

0001e 5f pop edi

0001f 5e pop esi

00020 5b pop ebx

00021 8b e5 mov esp, ebp

00023 5d pop ebp

00024 c3 ret 0

??$\_\_vcrt\_va\_start\_verify\_argument\_type@QBD@@YAXXZ ENDP ; \_\_vcrt\_va\_start\_verify\_argument\_type<char const \* const>

\_TEXT ENDS

; Function compile flags: /Odtp /RTCsu /ZI

; File c:\program files (x86)\windows kits\10\include\10.0.10150.0\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

; 638 : {

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

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

0001e 8b f4 mov esi, esp

00020 8b 45 14 mov eax, DWORD PTR \_\_ArgList$[ebp]

00023 50 push eax

00024 8b 4d 10 mov ecx, DWORD PTR \_\_Locale$[ebp]

00027 51 push ecx

00028 8b 55 0c mov edx, DWORD PTR \_\_Format$[ebp]

0002b 52 push edx

0002c 8b 45 08 mov eax, DWORD PTR \_\_Stream$[ebp]

0002f 50 push eax

00030 e8 00 00 00 00 call \_\_\_local\_stdio\_printf\_options

00035 8b 48 04 mov ecx, DWORD PTR [eax+4]

00038 51 push ecx

00039 8b 10 mov edx, DWORD PTR [eax]

0003b 52 push edx

0003c ff 15 00 00 00

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

00042 83 c4 18 add esp, 24 ; 00000018H

00045 3b f4 cmp esi, esp

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

; 640 : }

0004c 5f pop edi

0004d 5e pop esi

0004e 5b pop ebx

0004f 81 c4 c0 00 00

00 add esp, 192 ; 000000c0H

00055 3b ec cmp ebp, esp

00057 e8 00 00 00 00 call \_\_RTC\_CheckEsp

0005c 8b e5 mov esp, ebp

0005e 5d pop ebp

0005f c3 ret 0

\_\_vfprintf\_l ENDP

\_TEXT ENDS

; Function compile flags: /Odtp /RTCsu /ZI

; File c:\program files (x86)\windows kits\10\include\10.0.10150.0\ucrt\corecrt\_stdio\_config.h

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

\_TEXT SEGMENT

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

; 73 : {

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

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

; 75 : return &\_OptionsStorage;

0001e b8 00 00 00 00 mov eax, OFFSET ?\_OptionsStorage@?1??\_\_local\_stdio\_printf\_options@@9@4\_KA ; `\_\_local\_stdio\_printf\_options'::`2'::\_OptionsStorage

; 76 : }

00023 5f pop edi

00024 5e pop esi

00025 5b pop ebx

00026 8b e5 mov esp, ebp

00028 5d pop ebp

00029 c3 ret 0

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

\_TEXT ENDS

END

**С++ програма з ассемблерними вставками**

#include <stdio.h>

int A[7];

int d;

int main() {

int i;

d = 1;

i = 0;

while (i <= 6) {

\_asm {

; A[i] = i | d;

mov eax, d

mov ecx, i

or eax, ecx

shl ecx,2

mov A[ecx], eax

;d += A[i];

mov eax, d

add eax, A[ecx]

mov d, eax

; if (d>10)

cmp eax, 10 ; d, 0000000aH

jle SHORT LN7main

; A[i] = 127 - i;

mov eax, 127 ; 0000007fH

sub eax, i

mov A[ecx], eax

LN7main:

; i++;

inc i

}

}

for (i = 0;i < 7;i++)

printf("%d ", A[i]);

printf("\n");

return 0;

}

**Лістинг №2**

; Listing generated by Microsoft (R) Optimizing Compiler Version 19.00.23026.0

TITLE D:\Projects\Lab2.2-5\Lab2.2-5\Lab2.2.cpp

.686P

.XMM

include listing.inc

.model flat

INCLUDELIB MSVCRTD

INCLUDELIB OLDNAMES

PUBLIC ?A@@3PAHA ; A

PUBLIC ?d@@3HA ; d

\_BSS SEGMENT

?A@@3PAHA DD 07H DUP (?) ; A

?d@@3HA DD 01H DUP (?) ; d

\_BSS ENDS

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

PUBLIC \_\_vfprintf\_l

PUBLIC ??$\_\_vcrt\_va\_start\_verify\_argument\_type@QBD@@YAXXZ ; \_\_vcrt\_va\_start\_verify\_argument\_type<char const \* const>

PUBLIC \_printf

PUBLIC \_main

PUBLIC ?\_OptionsStorage@?1??\_\_local\_stdio\_printf\_options@@9@4\_KA ; `\_\_local\_stdio\_printf\_options'::`2'::\_OptionsStorage

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

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

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

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

EXTRN @\_\_security\_check\_cookie@4:PROC

EXTRN \_\_RTC\_CheckEsp:PROC

EXTRN \_\_RTC\_InitBase:PROC

EXTRN \_\_RTC\_Shutdown:PROC

EXTRN \_\_\_security\_cookie:DWORD

; COMDAT ?\_OptionsStorage@?1??\_\_local\_stdio\_printf\_options@@9@4\_KA

\_BSS SEGMENT

?\_OptionsStorage@?1??\_\_local\_stdio\_printf\_options@@9@4\_KA DQ 01H DUP (?) ; `\_\_local\_stdio\_printf\_options'::`2'::\_OptionsStorage

\_BSS 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?$AA@

CONST SEGMENT

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

CONST ENDS

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

CONST SEGMENT

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

CONST ENDS

; Function compile flags: /Odtp /RTCsu /ZI

; File d:\projects\lab2.2-5\lab2.2-5\lab2.2.cpp

; COMDAT \_main

\_TEXT SEGMENT

\_i$ = -12 ; size = 4

\_\_$ArrayPad$ = -4 ; size = 4

\_main PROC ; COMDAT

; 4 : int main() {

00000 55 push ebp

00001 8b ec mov ebp, esp

00003 81 ec d0 00 00

00 sub esp, 208 ; 000000d0H

00009 53 push ebx

0000a 56 push esi

0000b 57 push edi

0000c 8d bd 30 ff ff

ff lea edi, DWORD PTR [ebp-208]

00012 b9 34 00 00 00 mov ecx, 52 ; 00000034H

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

0001c f3 ab rep stosd

0001e a1 00 00 00 00 mov eax, DWORD PTR \_\_\_security\_cookie

00023 33 c5 xor eax, ebp

00025 89 45 fc mov DWORD PTR \_\_$ArrayPad$[ebp], eax

; 5 : int i;

; 6 : d = 1;

00028 c7 05 00 00 00

00 01 00 00 00 mov DWORD PTR ?d@@3HA, 1 ; d

; 7 : i = 0;

00032 c7 45 f4 00 00

00 00 mov DWORD PTR \_i$[ebp], 0

$LN2@main:

; 8 : while (i <= 6) {

00039 83 7d f4 06 cmp DWORD PTR \_i$[ebp], 6

0003d 7f 3b jg SHORT $LN3@main

; 9 : \_asm {

; 10 : ; A[i] = i | d;

; 11 : mov eax, d

0003f a1 00 00 00 00 mov eax, DWORD PTR ?d@@3HA ; d

; 12 : mov ecx, i

00044 8b 4d f4 mov ecx, DWORD PTR \_i$[ebp]

; 13 : or eax, ecx

00047 0b c1 or eax, ecx

; 14 : shl ecx,2

00049 c1 e1 02 shl ecx, 2

; 15 : mov A[ecx], eax

0004c 89 81 00 00 00

00 mov DWORD PTR ?A@@3PAHA[ecx], eax

; 16 :

; 17 : ;d += A[i];

; 18 : mov eax, d

00052 a1 00 00 00 00 mov eax, DWORD PTR ?d@@3HA ; d

; 19 : add eax, A[ecx]

00057 03 81 00 00 00

00 add eax, DWORD PTR ?A@@3PAHA[ecx]

; 20 : mov d, eax

0005d a3 00 00 00 00 mov DWORD PTR ?d@@3HA, eax ; d

; 21 :

; 22 : ; if (d>10)

; 23 : cmp eax, 10 ; d, 0000000aH

00062 83 f8 0a cmp eax, 10 ; 0000000aH

; 24 : jle SHORT LN7main

00065 7e 0e jle SHORT $LN7main$8

; 25 :

; 26 : ; A[i] = 127 - i;

; 27 : mov eax, 127 ; 0000007fH

00067 b8 7f 00 00 00 mov eax, 127 ; 0000007fH

; 28 : sub eax, i

0006c 2b 45 f4 sub eax, DWORD PTR \_i$[ebp]

; 29 : mov A[ecx], eax

0006f 89 81 00 00 00

00 mov DWORD PTR ?A@@3PAHA[ecx], eax

$LN7main$8:

; 30 : LN7main:

; 31 : ; i++;

; 32 : inc i

00075 ff 45 f4 inc DWORD PTR \_i$[ebp]

; 33 :

; 34 : }

; 35 : }

00078 eb bf jmp SHORT $LN2@main

$LN3@main:

; 36 : for (i = 0;i < 7;i++)

0007a c7 45 f4 00 00

00 00 mov DWORD PTR \_i$[ebp], 0

00081 eb 09 jmp SHORT $LN6@main

$LN4@main:

00083 8b 45 f4 mov eax, DWORD PTR \_i$[ebp]

00086 83 c0 01 add eax, 1

00089 89 45 f4 mov DWORD PTR \_i$[ebp], eax

$LN6@main:

0008c 83 7d f4 07 cmp DWORD PTR \_i$[ebp], 7

00090 7d 1a jge SHORT $LN5@main

; 37 : printf("%d ", A[i]);

00092 8b 45 f4 mov eax, DWORD PTR \_i$[ebp]

00095 8b 0c 85 00 00

00 00 mov ecx, DWORD PTR ?A@@3PAHA[eax\*4]

0009c 51 push ecx

0009d 68 00 00 00 00 push OFFSET ??\_C@\_03JDANDILB@?$CFd?5?$AA@

000a2 e8 00 00 00 00 call \_printf

000a7 83 c4 08 add esp, 8

000aa eb d7 jmp SHORT $LN4@main

$LN5@main:

; 38 : printf("\n");

000ac 68 00 00 00 00 push OFFSET ??\_C@\_01EEMJAFIK@?6?$AA@

000b1 e8 00 00 00 00 call \_printf

000b6 83 c4 04 add esp, 4

; 39 : return 0;

000b9 33 c0 xor eax, eax

; 40 : }

000bb 5f pop edi

000bc 5e pop esi

000bd 5b pop ebx

000be 8b 4d fc mov ecx, DWORD PTR \_\_$ArrayPad$[ebp]

000c1 33 cd xor ecx, ebp

000c3 e8 00 00 00 00 call @\_\_security\_check\_cookie@4

000c8 81 c4 d0 00 00

00 add esp, 208 ; 000000d0H

000ce 3b ec cmp ebp, esp

000d0 e8 00 00 00 00 call \_\_RTC\_CheckEsp

000d5 8b e5 mov esp, ebp

000d7 5d pop ebp

000d8 c3 ret 0

\_main ENDP

\_TEXT ENDS

; Function compile flags: /Odtp /RTCsu /ZI

; File c:\program files (x86)\windows kits\10\include\10.0.10150.0\ucrt\stdio.h

; COMDAT \_printf

\_TEXT SEGMENT

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

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

\_\_Format$ = 8 ; size = 4

\_printf PROC ; COMDAT

; 950 : {

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

; 951 : int \_Result;

; 952 : va\_list \_ArgList;

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

0001e e8 00 00 00 00 call ??$\_\_vcrt\_va\_start\_verify\_argument\_type@QBD@@YAXXZ ; \_\_vcrt\_va\_start\_verify\_argument\_type<char const \* const>

00023 8d 45 0c lea eax, DWORD PTR \_\_Format$[ebp+4]

00026 89 45 ec mov DWORD PTR \_\_ArgList$[ebp], eax

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

00029 8b 45 ec mov eax, DWORD PTR \_\_ArgList$[ebp]

0002c 50 push eax

0002d 6a 00 push 0

0002f 8b 4d 08 mov ecx, DWORD PTR \_\_Format$[ebp]

00032 51 push ecx

00033 8b f4 mov esi, esp

00035 6a 01 push 1

00037 ff 15 00 00 00

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

0003d 83 c4 04 add esp, 4

00040 3b f4 cmp esi, esp

00042 e8 00 00 00 00 call \_\_RTC\_CheckEsp

00047 50 push eax

00048 e8 00 00 00 00 call \_\_vfprintf\_l

0004d 83 c4 10 add esp, 16 ; 00000010H

00050 89 45 f8 mov DWORD PTR \_\_Result$[ebp], eax

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

00053 c7 45 ec 00 00

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

; 956 : return \_Result;

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

; 957 : }

0005d 5f pop edi

0005e 5e pop esi

0005f 5b pop ebx

00060 81 c4 d8 00 00

00 add esp, 216 ; 000000d8H

00066 3b ec cmp ebp, esp

00068 e8 00 00 00 00 call \_\_RTC\_CheckEsp

0006d 8b e5 mov esp, ebp

0006f 5d pop ebp

00070 c3 ret 0

\_printf ENDP

\_TEXT ENDS

; Function compile flags: /Odtp /RTCsu /ZI

; File d:\program files (x86)\vs\vc\include\vadefs.h

; COMDAT ??$\_\_vcrt\_va\_start\_verify\_argument\_type@QBD@@YAXXZ

\_TEXT SEGMENT

??$\_\_vcrt\_va\_start\_verify\_argument\_type@QBD@@YAXXZ PROC ; \_\_vcrt\_va\_start\_verify\_argument\_type<char const \* const>, COMDAT

; 144 : {

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

; 145 : static\_assert(!\_\_vcrt\_va\_list\_is\_reference<\_Ty>::\_\_the\_value, "va\_start argument must not have reference type and must not be parenthesized");

; 146 : }

0001e 5f pop edi

0001f 5e pop esi

00020 5b pop ebx

00021 8b e5 mov esp, ebp

00023 5d pop ebp

00024 c3 ret 0

??$\_\_vcrt\_va\_start\_verify\_argument\_type@QBD@@YAXXZ ENDP ; \_\_vcrt\_va\_start\_verify\_argument\_type<char const \* const>

\_TEXT ENDS

; Function compile flags: /Odtp /RTCsu /ZI

; File c:\program files (x86)\windows kits\10\include\10.0.10150.0\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

; 638 : {

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

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

0001e 8b f4 mov esi, esp

00020 8b 45 14 mov eax, DWORD PTR \_\_ArgList$[ebp]

00023 50 push eax

00024 8b 4d 10 mov ecx, DWORD PTR \_\_Locale$[ebp]

00027 51 push ecx

00028 8b 55 0c mov edx, DWORD PTR \_\_Format$[ebp]

0002b 52 push edx

0002c 8b 45 08 mov eax, DWORD PTR \_\_Stream$[ebp]

0002f 50 push eax

00030 e8 00 00 00 00 call \_\_\_local\_stdio\_printf\_options

00035 8b 48 04 mov ecx, DWORD PTR [eax+4]

00038 51 push ecx

00039 8b 10 mov edx, DWORD PTR [eax]

0003b 52 push edx

0003c ff 15 00 00 00

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

00042 83 c4 18 add esp, 24 ; 00000018H

00045 3b f4 cmp esi, esp

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

; 640 : }

0004c 5f pop edi

0004d 5e pop esi

0004e 5b pop ebx

0004f 81 c4 c0 00 00

00 add esp, 192 ; 000000c0H

00055 3b ec cmp ebp, esp

00057 e8 00 00 00 00 call \_\_RTC\_CheckEsp

0005c 8b e5 mov esp, ebp

0005e 5d pop ebp

0005f c3 ret 0

\_\_vfprintf\_l ENDP

\_TEXT ENDS

; Function compile flags: /Odtp /RTCsu /ZI

; File c:\program files (x86)\windows kits\10\include\10.0.10150.0\ucrt\corecrt\_stdio\_config.h

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

\_TEXT SEGMENT

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

; 73 : {

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

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

; 75 : return &\_OptionsStorage;

0001e b8 00 00 00 00 mov eax, OFFSET ?\_OptionsStorage@?1??\_\_local\_stdio\_printf\_options@@9@4\_KA ; `\_\_local\_stdio\_printf\_options'::`2'::\_OptionsStorage

; 76 : }

00023 5f pop edi

00024 5e pop esi

00025 5b pop ebx

00026 8b e5 mov esp, ebp

00028 5d pop ebp

00029 c3 ret 0

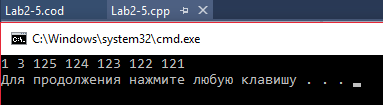
\_\_\_local\_stdio\_printf\_options ENDP

\_TEXT ENDS

END

**Порівняння результатів**

С++ програма



С++ програма з ассемблерними вставками

