НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Системне програмування »

*Звiт з лабораторної роботи №6*

*Тема: «*SSE- та AVX-команди, графічні фігури*»*

Виконав:

ст. гр. KIT-119D

Гряник Г.В.

Перевірив:

Межерицький С.Г.

Харків – 2020

**Мета :**

набути практичних навичок складання, налагодження та виконання програм з використанням команд SSE і AVX для платформи х64 в середовищі masm64, з віконним інтерфейсом для програмування арифметичних операцій за допомогою SSE (AVX) команд під МП.

***Індивідуальне завдання***

Варіант 6.

* Виконати операцію паралельного логічного множення 2-х масивів по 10-ть 64-розрядних дійсних чисел. Скласти все результати. Якщо сума всіх отриманих чисел більше 100, то виконати обчислення квадратного кореня, а якщо навпаки - не виконувати.
* Малювати циклоїдальних криву у вигляді гіпоціклоіди по параметричних рівнянь:x = (R - mR) cos (mt) + mcos (t - mt); y = (R - mR) sin (mt) - msin (t - mt), при m = 1/4, 1 / 2.1,де R - радіус нерухомою кола; r - радіус кола, що котиться; m = r / R.

**Текст програми**

Програма 1: (LR6-1)

include \masm64\include64\masm64rt.inc ; библиотеки

.data ;

mas1 dd 3.,1.,11.,1.,0.,5.,15.,1.,0.,9.,0.,0. ; массив чисел arr1 размером в слово

len1 equ ($-mas1)/type mas1 ; количество чисел массива

mas2 dd 5., 3.,0.,4.,83.,0.,2.,3.,4.,3.,0.,0. ; массив чисел arr2 размером в слово

mas1\_2 dd len1 dup(0) ; количество чисел массива результата

\_res dd 0. ; операнд res1 размерностью 64 разряда; змінна результ

a DD 100.0 ;

;Текст для MessageBox

title1 db "Лаб.6-1 SSE and AVX masm64",0

buf1 dq 55 dup(0),0

buf2 dq 25 dup(0),0

buf3 dq 25 dup(0),0

buf4 dq 25 dup(0),0

;Текст для window

txt01 db "Виконати операцію паралельного логічного множення 2-х масивів по 10-ть 64-розрядних дійсних чисел. Скласти всi результати. Якщо сума всіх отриманих чисел більше 100, то виконати обчислення квадратного кореня, а якщо навпаки - не виконувати.",0

txt02 db "Сума всіx чисел: %s",0

txt03 db "Результат обчислення квадратного кореня: %d",0

txt04 db "За умовою корінь добувати неможна",0

autor db "Автор: Гряник Г.В., гр.КІТ-119Д",0

txt05 db "Більше 100 ",0

txt06 db "Менше 100",0

txt07 db "Корінь: %d",0

hInstance dq ? ; дескриптор програми

hWnd dq ? ; дескриптор окна

hIcon dq ? ; дескриптор иконки

hBmp dq ?

hStatic dq ?

classname db "template\_class",0

;вихідний файл

fNameS BYTE "Res\_LR6-1S.txt",0;файл для зберігання

fNameA BYTE "Res\_LR6-1A.txt",0;файл для зберігання

fHandle dq ? ; для даних файлу

cWritten dq ? ;

BSIZE equ 27;розмір файлу

.code;cекция кода

entry\_point proc

lea rsi, mas1 ; занесення адреси масиву mas1 в rsi

lea rdi, mas2 ; занесення адреси масиву mas2 в rdi

lea rbx, mas1\_2

movups XMM0,[rsi] ;mas1 ;

movups XMM1,[rdi] ;mas2 ;

mulps xmm1,xmm0

;movupd xmmword ptr mas1\_2,xmm1

movups XMM2,[rsi+16] ;mas1 ;

movups XMM3,[rdi+16] ;mas2 ;

mulps xmm3,xmm2

movups XMM4,[rsi+32] ;mas1 ;

movups XMM5,[rdi+32] ;mas2 ;

mulps xmm5,xmm4

addps xmm1,xmm3

addps xmm1,xmm5

movupd xmmword ptr mas1\_2,xmm1

movups XMM0,mas1\_2

movaps XMM1,XMM0 ;

shufps XMM1,XMM1,11111001b ;

addss XMM0,XMM1 ;

shufps XMM1,XMM1,11111001b ;

addss XMM0,XMM1 ;

shufps XMM1,XMM1,11111001b ;

addss XMM0,XMM1 ;

movupd xmmword ptr a,xmm0

cvttss2si eax,xmm0

movsxd r15,eax

invoke wsprintf,ADDR buf2,ADDR txt02,ADDR txt06

invoke wsprintf,ADDR buf4,ADDR txt04

cmp r15,100

JNGE EXIT

;mov a, eax

lea r15,a

MOVUPS XMM5, a

SQRTSS xmm6, xmm5

cvttss2si rdi,xmm6

invoke wsprintf,ADDR buf2,ADDR txt02, ADDR txt05

invoke wsprintf,ADDR buf4,ADDR txt07, rdi

EXIT:

;Створення Message

invoke wsprintf,ADDR buf1,ADDR txt01

invoke wsprintf,ADDR buf3,ADDR txt03,r15

mov hInstance,rv(GetModuleHandle,0)

mov hIcon,rv(LoadImage,hInstance,10,IMAGE\_ICON,128,128,LR\_DEFAULTCOLOR)

mov hBmp, rv(ResImageLoad,20,,128,128,LR\_DEFAULTCOLOR)

invoke DialogBoxParam,hInstance,1000,0,ADDR main,hIcon

invoke ExitProcess,0

ret

entry\_point endp

main proc hWin:QWORD,uMsg:QWORD,wParam:QWORD,lParam:QWORD

.switch uMsg

.case WM\_INITDIALOG ; сообщение о создании диал. окна

invoke SetWindowText,hWin,title1

invoke SendMessage,hWin,WM\_SETICON,1,lParam ; отправляет сообщение окну

invoke SendMessage,rv(GetDlgItem,hWin,101),\ ; сообщение окну по дескриптору органа управления

STM\_SETIMAGE,IMAGE\_ICON,lParam

; 102 - jpg

mov hStatic, rv(GetDlgItem,hWin,101)

invoke SendMessage,hStatic,STM\_SETIMAGE,IMAGE\_BITMAP,hBmp

.return TRUE

.case WM\_COMMAND ; сообщение от меню или кнопки

.switch wParam

.case 103 ; кнопка

invoke MsgboxI,hWin,ADDR buf1,"Завдання",MB\_OK,10

;rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0; закрытие окна

.case 105

invoke MsgboxI,hWin,ADDR autor,"Автор",MB\_OK,10

.case 106

invoke MsgboxI,hWin,ADDR buf2,"Результат:",MB\_OK,10

invoke MsgboxI,hWin,ADDR buf4,"Корінь:",MB\_OK,10

.case 107

invoke CreateFile,ADDR fNameS,GENERIC\_WRITE,0,0,CREATE\_ALWAYS,FILE\_ATTRIBUTE\_ARCHIVE,0

mov fHandle, rax;відкрити та записа данні файла

invoke WriteFile,fHandle,ADDR buf2 ,BSIZE ,ADDR cWritten,0

invoke WriteFile,fHandle,ADDR buf3 ,BSIZE ,ADDR cWritten,0

invoke MsgboxI,hWin,"Успішно збережено","SAVE",MB\_OK,10

.case 110 ; кнопка

invoke MsgboxI,hWin,ADDR buf1,"Завдання",MB\_OK,10

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0; закрытие окна

.case 113

invoke MsgboxI,hWin,ADDR autor,"Автор",MB\_OK,10

.case 112

invoke MsgboxI,hWin,ADDR buf2,"Результат:",MB\_OK,10

.case 116

invoke MsgboxI,hWin,ADDR buf4,"Корінь:",MB\_OK,10

.case 115

invoke CreateFile,ADDR fNameS,GENERIC\_WRITE,0,0,CREATE\_ALWAYS,FILE\_ATTRIBUTE\_ARCHIVE,0

mov fHandle, rax;відкрити та записа данні файла

invoke WriteFile,fHandle,ADDR buf1 ,BSIZE ,ADDR cWritten,0

invoke WriteFile,fHandle,ADDR buf2 ,BSIZE ,ADDR cWritten,0

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0;

.case 117

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0;

.endsw

.case WM\_CLOSE ; если есть сообщение о закрытии окна

invoke EndDialog,hWin,0 ;

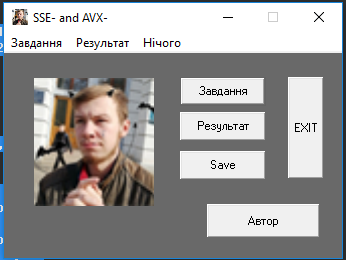
.endsw

xor rax, rax

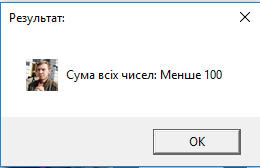
ret

main endp

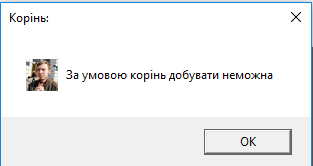
end



А)

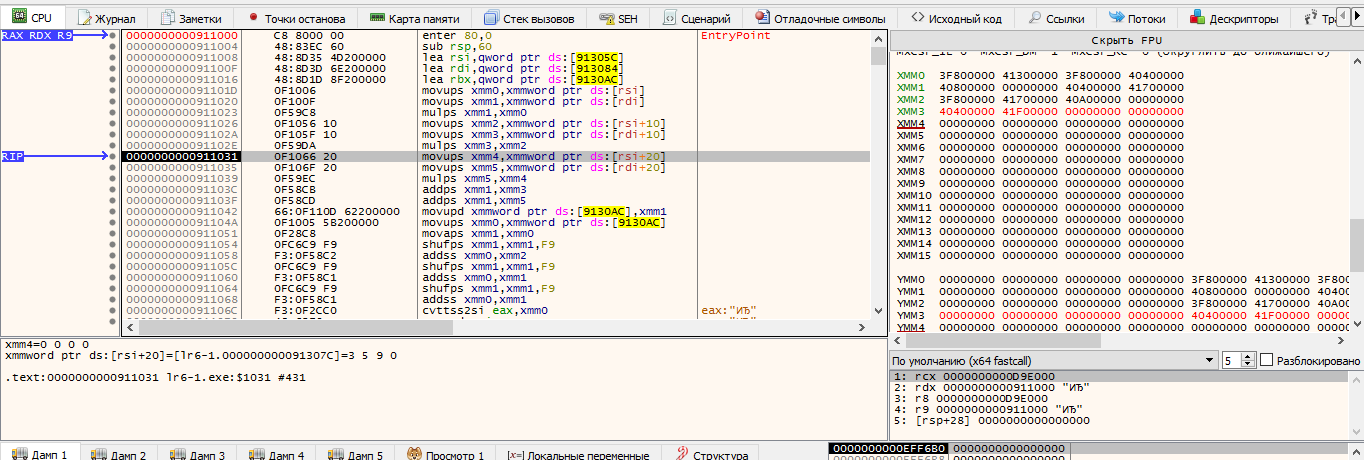


Б)



В)

**Рис 3 – результат виконання роботи програми** LR6-1



**Рис 4 – результат налагодження програми** LR6-1

Алгоритм: Задано два масива. Значення цих масивів заносяться до xmm регістру після знаходиться добуток двох масивів та результат записується в третій масив. Далі знаходиться сума елементів результуючого масиву і якщо сума більше 100 то програма виводить відповідне повідомленні та знаходить корінь від результату в іншому випадку програма виводить повідомлення про те що сума менше 100 та добуток кореня заборонено. Вивід всіх даних програми реалізовано завдяки окремому створеному вікну.

**Програма 2**: (LR6-1.2)

include \masm64\include64\masm64rt.inc ; библиотеки

.data ;

mas1 dd 3.,1.,11.,1.,0.,5.,15.,1.,0.,9.,0.,0.0 ; массив чисел arr1 размером в слово

len1 equ ($-mas1)/type mas1 ; количество чисел массива

mas1\_2 dd len1 dup(0)

mas2 dd 5., 3.,0.,4.,83.,0.,2.,3.,4.,3.,0.,0. ; массив чисел arr2 размером в слово

mas2\_2 dd len1 dup(0) ; количество чисел массива результата

\_res dd 0. ; операнд res1 размерностью 64 разряда; змінна результ

a DD 82.0 ;

;Текст для MessageBox

title1 db "Лаб.6-1 SSE and AVX masm64",0

buf1 dq 55 dup(0),0

buf2 dq 25 dup(0),0

buf3 dq 25 dup(0),0

buf4 dq 25 dup(0),0

;Текст для window

txt01 db "Виконати операцію паралельного логічного множення 2-х масивів по 10-ть 64-розрядних дійсних чисел. Скласти всi результати. Якщо сума всіх отриманих чисел більше 100, то виконати обчислення квадратного кореня, а якщо навпаки - не виконувати.",0

txt02 db "Сума всіx чисел: %s",0

txt03 db "Результат обчислення квадратного кореня: %d",0

txt04 db "За умовою корінь добувати неможна",0

autor db "Автор: Гряник Г.В., гр.КІТ-119Д",0

txt05 db "Більше 100",0

txt06 db "Менше 100",0

txt07 db "Корінь: %d",0

hInstance dq ? ; дескриптор програми

hWnd dq ? ; дескриптор окна

hIcon dq ? ; дескриптор иконки

hBmp dq ?

hStatic dq ?

classname db "template\_class",0

;вихідний файл

fNameS BYTE "Res\_LR6-1S.txt",0;файл для зберігання

fNameA BYTE "Res\_LR6-1A.txt",0;файл для зберігання

fHandle dq ? ; для даних файлу

cWritten dq ? ;

BSIZE equ 27;розмір файлу

.code;cекция кода

entry\_point proc

lea rsi, mas1 ; занесення адреси масиву mas1 в rsi

lea rdi, mas2 ; занесення адреси масиву mas2 в rdi

lea rbx, mas1\_2

m1: vmovups ymm0,[rsi] ; перемещение упакованных чисел одинарной точности

vmovups ymm1,[rdi] ;

vmulps ymm2,ymm0,ymm1;

vmovups [rbx],ymm2 ; перемещение в память по адресу регистра rbx

add rdi,32 ; 32 х 8 = 256

add rsi,32 ; смещение на 256

add rbx,32 ; смещение на 32 байта = 256

mov rcx,rdx ; занесение остатка в счетчик

vmovss xmm3,dword ptr [rsi+4] ; занесение элемента mas1

vmulps XMM3,XMM3,dword ptr [rdi+4] ; сложение с элементом mas2

vmovss dword ptr [rbx+4],XMM3 ; сохранение в r1

vmovss xmm3,dword ptr [rsi+8] ; занесение элемента mas1

vmulps XMM3,XMM3,dword ptr [rdi+8] ; сложение с элементом mas2

vmovss dword ptr [rbx+8],XMM3 ; сохранение в r1

lea rbx, mas1\_2

vmovss xmm3, dword ptr [rbx] ;

vaddss XMM3,XMM3,dword ptr [rbx+4]

vaddss XMM3,XMM3,dword ptr [rbx+8]

vaddss XMM3,XMM3,dword ptr [rbx+12]

vaddss XMM3,XMM3,dword ptr [rbx+16]

vaddss XMM3,XMM3,dword ptr [rbx+20]

vaddss XMM3,XMM3,dword ptr [rbx+24]

vaddss XMM3,XMM3,dword ptr [rbx+28]

vaddss XMM3,XMM3,dword ptr [rbx+32]

vaddss XMM3,XMM3,dword ptr [rbx+36]

vaddss XMM3,XMM3,dword ptr [rbx+40]

vmovss dword ptr a,XMM3

invoke wsprintf,ADDR buf2,ADDR txt02,ADDR txt06

invoke wsprintf,ADDR buf4, ADDR txt04

mov r15,rsi

cmp a,100

;JNGE EXIT

;mov a, eax

MOVUPS XMM5, a

SQRTSS xmm6, xmm5

cvttss2si rdi,xmm6

invoke wsprintf,ADDR buf2,ADDR txt02, ADDR txt05

invoke wsprintf,ADDR buf4,ADDR txt07, rdi

EXIT:

;Створення Message

invoke wsprintf,ADDR buf1,ADDR txt01

invoke wsprintf,ADDR buf3,ADDR txt03,r15

mov hInstance,rv(GetModuleHandle,0)

mov hIcon,rv(LoadImage,hInstance,10,IMAGE\_ICON,128,128,LR\_DEFAULTCOLOR)

mov hBmp, rv(ResImageLoad,20,,128,128,LR\_DEFAULTCOLOR)

invoke DialogBoxParam,hInstance,1000,0,ADDR main,hIcon

invoke ExitProcess,0

ret

entry\_point endp

main proc hWin:QWORD,uMsg:QWORD,wParam:QWORD,lParam:QWORD

.switch uMsg

.case WM\_INITDIALOG ; сообщение о создании диал. окна

invoke SetWindowText,hWin,title1

invoke SendMessage,hWin,WM\_SETICON,1,lParam ; отправляет сообщение окну

invoke SendMessage,rv(GetDlgItem,hWin,101),\ ; сообщение окну по дескриптору органа управления

STM\_SETIMAGE,IMAGE\_ICON,lParam

; 102 - jpg

mov hStatic, rv(GetDlgItem,hWin,101)

invoke SendMessage,hStatic,STM\_SETIMAGE,IMAGE\_BITMAP,hBmp

.return TRUE

.case WM\_COMMAND ; сообщение от меню или кнопки

.switch wParam

.case 103 ; кнопка

invoke MsgboxI,hWin,ADDR buf1,"Завдання",MB\_OK,10

;rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0; закрытие окна

.case 105

invoke MsgboxI,hWin,ADDR autor,"Автор",MB\_OK,10

.case 106

invoke MsgboxI,hWin,ADDR buf2,"Результат:",MB\_OK,10

invoke MsgboxI,hWin,ADDR buf4,"Корінь:",MB\_OK,10

.case 107

invoke CreateFile,ADDR fNameS,GENERIC\_WRITE,0,0,CREATE\_ALWAYS,FILE\_ATTRIBUTE\_ARCHIVE,0

mov fHandle, rax;відкрити та записа данні файла

invoke WriteFile,fHandle,ADDR buf2 ,BSIZE ,ADDR cWritten,0

invoke WriteFile,fHandle,ADDR buf3 ,BSIZE ,ADDR cWritten,0

invoke MsgboxI,hWin,"Успішно збережено","SAVE",MB\_OK,10

.case 110 ; кнопка

invoke MsgboxI,hWin,ADDR buf1,"Завдання",MB\_OK,10

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0; закрытие окна

.case 113

invoke MsgboxI,hWin,ADDR autor,"Автор",MB\_OK,10

.case 112

invoke MsgboxI,hWin,ADDR buf2,"Результат:",MB\_OK,10

.case 116

invoke MsgboxI,hWin,ADDR buf4,"Корінь:",MB\_OK,10

.case 115

invoke CreateFile,ADDR fNameS,GENERIC\_WRITE,0,0,CREATE\_ALWAYS,FILE\_ATTRIBUTE\_ARCHIVE,0

mov fHandle, rax;відкрити та записа данні файла

invoke WriteFile,fHandle,ADDR buf1 ,BSIZE ,ADDR cWritten,0

invoke WriteFile,fHandle,ADDR buf2 ,BSIZE ,ADDR cWritten,0

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0;

.case 117

rcall SendMessage,hWin,WM\_SYSCOMMAND,SC\_CLOSE,0;

.endsw

.case WM\_CLOSE ; если есть сообщение о закрытии окна

invoke EndDialog,hWin,0 ;

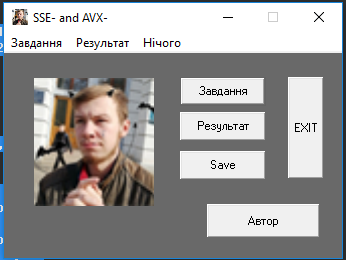
.endsw

xor rax, rax

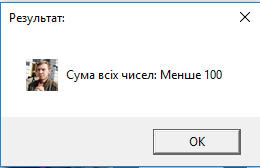
ret

main endp

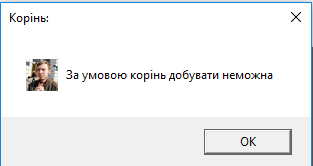
end



А)

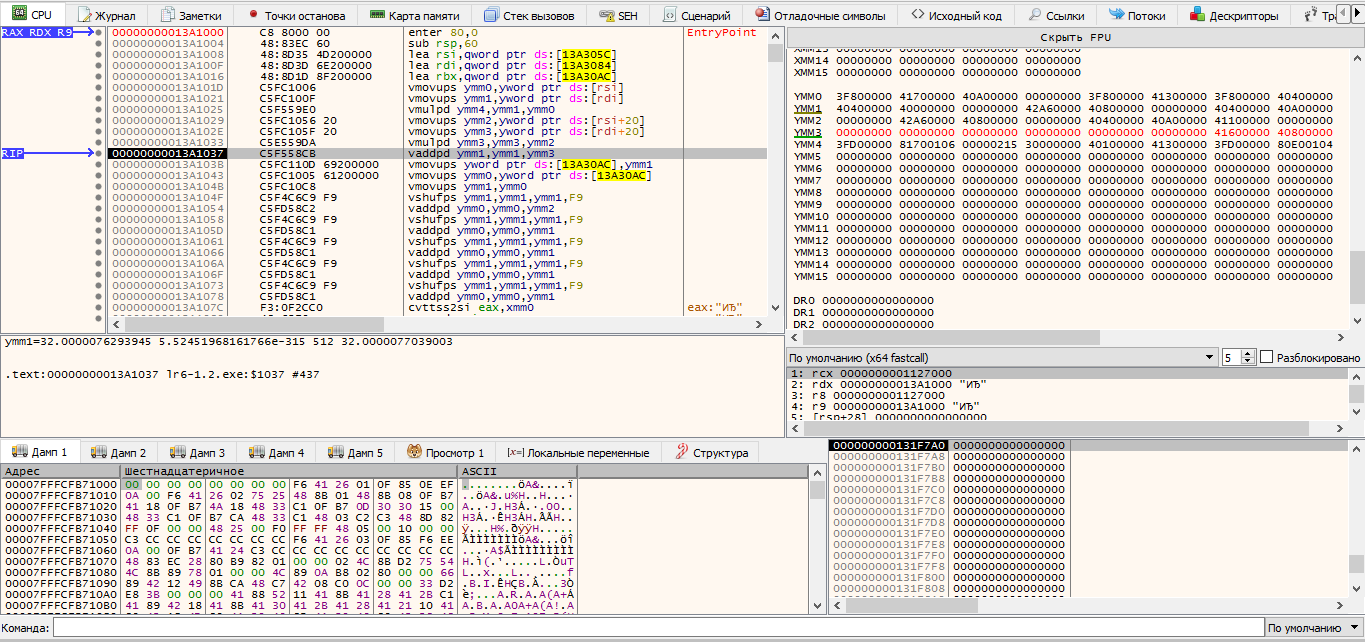


Б)



В)

**Рис 3 – результат виконання роботи програми** LR6-1.2



**Рис 4 – результат налагодження програми** LR6-1.2

Алгоритм: алгоритм ідентичний алгоритму попередньої програми за винятком використання yxx

**Програма 3:** (LR6-2)

include \masm64\include64\masm64rt.inc

.data

hInstance dq ? ; дескриптор програми

hWnd dq ? ; дескриптор окна

hIcon dq ? ; дескриптор иконки

hCursor dq ? ; дескриптор курсора

sWid dq ? ; ширина монитора (колич. пикселей по x)

sHgt dq ? ; высота монитора (колич. пикселей по y)

classname db "template\_class",0

caption db "Гряафічна фігура",0

MouseClick db 0 ; флаг нажатия

myXY POINT <> ; структура для х- та у-координат

mas dd 500 ;

m dd 0.9

R dd 1.0

r dd 1.0

t dd 0.0 ; угловая координата

delta dd 0.1175 ; один градус

xdiv2 dq ? ; середина по X

ydiv2 dq ? ; середина по Y

tmp dd 0 ; временная переменная

divK dd 100.0 ; масштабный коэффициент

xr dd 0. ; координаты функции

yr dd 0

temp1 dd 0

text01 db "x=%d y=%d",0

AppName db "D",0

buf1 dq 3 dup(0),0

.code

entry\_point proc

invoke GetSystemMetrics,SM\_CXSCREEN ; получение ширины экрана в пикселях

shr rax,1 ; деление на 2 – определение середины экрана по Х

mov xdiv2,rax

invoke GetSystemMetrics,SM\_CYSCREEN ; получение высоты экрана в пикселях

shr rax,1 ; деление на 2 – определение середины экрана по Y

mov ydiv2,rax

mov r10d,mas ; сохранение количества циклов

mov temp1,r10d

finit

l1: ; x = x0 + Кfcosf

fld m ; m

fmul t ; m\*alpha

fcos ;cos(m\*t)

fld R ; R

fld R ; R

Fmul m ; Rm

FSUB ; R-RM

FMUL ;(R-RM)cos(m\*t)

fld t ; t

fld t ; t

FMUL m ; R\*t

FSUB ; t-R\*t

FCOS ;cos(t-r\*t)

FMUL m ;m\*cos(t-rt)

FADD ;(R-RM)cos(m\*t)+m\*cos(t-rt)

FMUL divK ; розмір\*((R-RM)cos(m\*t)+m\*cos(t-rt))

fild xdiv2

fadd

fistp dword ptr xr

fld m

fmul t ; m\*alpha

fsin ;sin (m\*t)

fld R ; R

fld R ; R

Fmul m ; Rm

FSUB ; R-RM

FMUL ;(R-RM)sin (m\*t)

fld t ; t

fld t ; t

FMUL m ; R\*t

FSUB ; t-R\*t

FSIN ;sin (t-r\*t)

FMUL m ;m\*cos(t-rt)

FSUB ;(R-RM)sin (m\*t)-m\*sin (t-rt)

FMUL divK ; розмір\*((R-RM)sin (m\*t)+m\*sin (t-rt))FMUL divK

fstp tmp

fild ydiv2

fsub tmp

fistp dword ptr yr ; сохранение X для выведения на экран

invoke Sleep,1 ; задержка

invoke SetCursorPos,xr,yr ; установление курсора по xr, yr

movss XMM3,delta

addss XMM3,t

movss t,XMM3

dec temp1 ; уменьшение счетчика

jz l2 ; продолжение рисование

jmp l1 ; выход из цикла

l2:

mov hInstance,rv(GetModuleHandle,0) ; получение и сохранение дескрипторa програми

mov hIcon, rv(LoadIcon,hInstance,10) ; загрузка и сохранение дескрипторa иконки

mov hCursor,rv(LoadCursor,0,IDC\_ARROW) ; загрузка курсора и сохранение

mov sWid,rv(GetSystemMetrics,SM\_CXSCREEN) ; получение кол. пикселей по х монитора

mov sHgt,rv(GetSystemMetrics,SM\_CYSCREEN) ; получение кол. пикселей по y монитора

call main

invoke ExitProcess,0

ret

entry\_point endp

main proc

LOCAL wc :WNDCLASSEX ; объявление локальных переменных

LOCAL lft :QWORD ; Лок. переменные содержатся в стеке

LOCAL top :QWORD ; и существуют только во время вып. проц.

LOCAL wid :QWORD

LOCAL hgt :QWORD

mov wc.cbSize,SIZEOF WNDCLASSEX ; колич. байтов структуры

mov wc.style,CS\_BYTEALIGNCLIENT or CS\_BYTEALIGNWINDOW ; стиль окна

mov wc.lpfnWndProc,ptr$(WndProc) ; адрес процедуры WndProc

mov wc.cbClsExtra,0 ; количество байтов для структуры класса

mov wc.cbWndExtra,0 ; количество байтов для структуры окна

mrm wc.hInstance,hInstance ; заполнение полЯ дескриптора в структуре

mrm wc.hIcon, hIcon ; хэндл иконки

mrm wc.hCursor,hCursor ; хэндл курсора

mrm wc.hbrBackground,0 ; цвет окна

mov wc.lpszMenuName,0 ; заполнение поля в структуре с именем ресурса меню

mov wc.lpszClassName,ptr$(classname) ; имя класса

mrm wc.hIconSm,hIcon

invoke RegisterClassEx,ADDR wc ; регистрациЯ класса окна

mov wid, 520 ; ширина пользовательского окна в пикселях

mov hgt, 520 ; высота пользовательского окна в пикселях

mov rax,sWid ; колич. пикселей монитора по x

sub rax,wid ; дельта • = •(монитора) - х(окна пользователя)

shr rax,1 ; получение середины •

mov lft,rax ;

mov rax, sHgt ; колич. пикселей монитора по y

sub rax, hgt ;

shr rax, 1 ;

mov top, rax ;

invoke CreateWindowEx,WS\_EX\_LEFT or WS\_EX\_ACCEPTFILES, \

ADDR classname,ADDR caption, \

WS\_OVERLAPPED or WS\_VISIBLE or WS\_SYSMENU,\

lft,top,wid,hgt,0,0,hInstance,0

mov hWnd,rax ; сохранение дескриптора окна

call msgloop

ret

main endp

msgloop proc

LOCAL msg :MSG

LOCAL pmsg :QWORD

mov pmsg,ptr$(msg) ; получение адреса структуры сообщениЯ

jmp gmsg ; jump directly to GetMessage()

mloop:

invoke TranslateMessage,pmsg

invoke DispatchMessage,pmsg ; отправка на обслуживание к WndProc

gmsg:

test rax, rv(GetMessage,pmsg,0,0,0) ; пока GetMessage не вернет ноль

jnz mloop

ret

msgloop endp

WndProc proc hWin:QWORD,uMsg:QWORD,wParam:QWORD,lParam:QWORD

LOCAL hdc:HDC ; резервирование стека для дескриптора окна

LOCAL ps:PAINTSTRUCT ; для структуры PAINTSTRUCT

LOCAL rect:RECT ; для структуры координат RECT

.switch uMsg

.case WM\_DESTROY ; если есть сообщение про уничтожение окна

invoke PostQuitMessage,NULL

.case WM\_LBUTTONDOWN ; сообщение от левой клавиши

mov rax,lParam ; 32-разрядные координаты курсора мышки

and rax,0ffffh ; выделение младшей части - координаты Х

mov myXY.x,eax ; сохранение координаты X

mov rax,lParam ; 32-разрядные координаты курсора мышки

shr rax,16 ; сдвиг вправо на 4 байта для выделения Y

mov myXY.y,eax ; сохранение координаты Y

mov MouseClick,TRUE ; подтверждение получения координат

invoke InvalidateRect,hWnd,0,TRUE ; для перерисования окна

.case WM\_PAINT ; если есть сообщение о перерисовании

invoke BeginPaint,hWnd,ADDR ps ; заполнение структуры

mov hdc,rax ; сохранение контекста

; проверка установления флажка

mov r10d,mas ; сохранение количества циклов

mov temp1,r10d

invoke InvalidateRect,hWnd,0,TRUE ; вызов функции и WM\_PAINT

finit

invoke BeginPaint,hWnd, ADDR ps ; вызов подготовительной процедуры

mov hdc,rax ; сохранение контекста

l1:

invoke wsprintf,ADDR buf1,ADDR text01,xr,yr

invoke TextOut,hdc, 10,30,ADDR buf1,10

sub xr, 480

sub yr, 200

invoke TextOut,hdc,xr,yr,\ ; координаты начала текста

addr AppName ,1; адрес хранения текста и кол. байтов текста

fld m ; m

fmul t ; m\*alpha

fcos ;cos(m\*t)

fld R ; R

fld R ; R

Fmul m ; Rm

FSUB ; R-RM

FMUL ;(R-RM)cos(m\*t)

fld t ; t

fld t ; t

FMUL m ; R\*t

FSUB ; t-R\*t

FCOS ;cos(t-r\*t)

FMUL m ;m\*cos(t-rt)

FADD ;(R-RM)cos(m\*t)+m\*cos(t-rt)

FMUL divK ; розмір\*((R-RM)cos(m\*t)+m\*cos(t-rt))

fild xdiv2

fadd

fistp dword ptr xr

fld m

fmul t ; m\*alpha

fsin ;sin (m\*t)

fld R ; R

fld R ; R

Fmul m ; Rm

FSUB ; R-RM

FMUL ;(R-RM)sin (m\*t)

fld t ; t

fld t ; t

FMUL m ; R\*t

FSUB ; t-R\*t

FSIN ;sin (t-r\*t)

FMUL m ;m\*cos(t-rt)

FSUB ;(R-RM)sin (m\*t)-m\*sin (t-rt)

FMUL divK ; розмір\*((R-RM)sin (m\*t)+m\*sin (t-rt))FMUL divK

fstp tmp

fild ydiv2

fsub tmp

fistp dword ptr yr ; сохранение X для выведения на экран

invoke Sleep,10 ; задержка

invoke SetCursorPos,xr,yr ; установление курсора по xr, yr

movss XMM3,delta

addss XMM3,t

movss t,XMM3

dec temp1 ; уменьшение счетчика

jz l2 ; продолжение рисование

jmp l1 ; выход из цикла

l2:

invoke EndPaint,hWnd, ADDR ps ; количество байтов текста

.endif

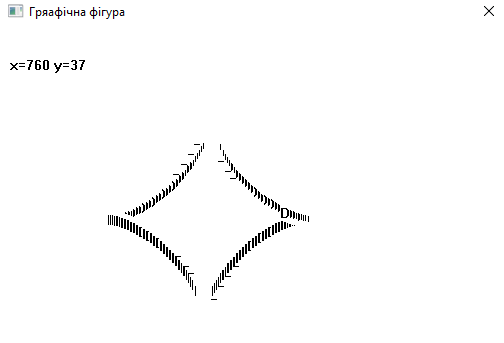
invoke DefWindowProc,hWin,uMsg,wParam,lParam

ret

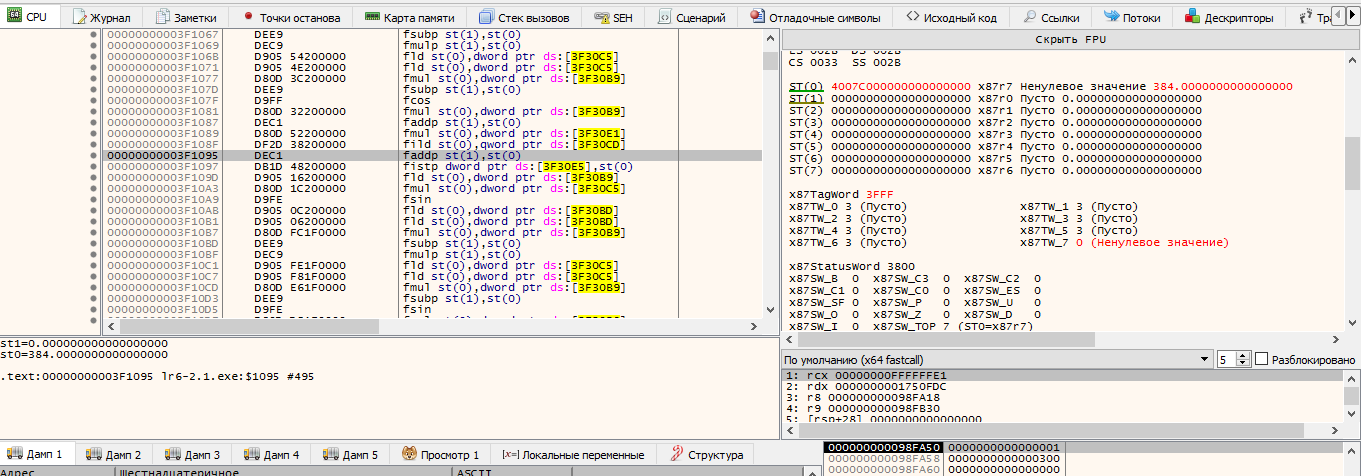
WndProc endp

End

**Результат**



**Рис 5 – результат виконання роботи програми** LR6-2



**Рис 6 – результат налагодження програми** LR6-2

**Алгоритм:** програма вирішує два вирази x = (R - mR) cos (mt) + mcos (t - mt); y = (R - mR) sin (mt) - msin (t - mt) і передає дані у відповідну функцію аби перемістити курсор. Це все робиться в нескінченному циклі. Паралельно у відкритому вікні малюється фігура яку вимальовує курсор та виводиться координати курсору.

**Висновок:** Під час лабораторної роботи було набуто практичних навичок з роботи mas64, а саме , навичок складання, налагодження та виконання програм з використанням рядкових команд, налагодження та виконання програм з використанням команд SSE і AVX для платформи х64 в середовищі masm64, з віконним інтерфейсом для програмування арифметичних операцій за допомогою SSE (AVX) команд під МП.