**Міністерство освіти і науки України**

**Національний технічний університет України**

**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

Лабораторна робота № 7

Модульне програмування. Використання процедур**.**

З дисципліни

Системне програмування

Виконав: Перевірив:

cтудент групи ІП-03 доц. кафедри ОТ

Янишин Віталій Павлов В.Г

Номер залікової книжки: 0327

Номер у списку групи: 21

Київ 2022

**Мета лабораторної роботи:** Вивчення прийомів модульного програмування, методів звернення до процедур і передачі в них параметрів.

**Хід роботи**

Варіант:

|  |  |
| --- | --- |
| 21 |  |

ОДЗ

1. 1+b\*c != 0 => b\*c !=-1 => b != -1/c
2. 53/a>=0 => a>0

Підібрані змінні (a, d, c)

A = { 0.5, 1.5, -3.5, 4.3, 42.1 }

B = { 2.3, 3.3, 5.1, 1,5, 5,5 }

C = { 5.6, -1.1, 2.2, -13.2, 1.5 }

D = { -11.1, 2.3, 12.22, -14.65, 1,3 }

Контрольні розрахунки

( sqrt2(53/1a) +4 d-54\*3b ) /8 ( 1+7b\*6c)

Цифри після арифметичних знаків символізують порядок виконання дій

Перша підстановка:

( sqrt(53/0.5) + (-11.1) – 4\*2.3) / (1+ 2.3\*5.6)

1. 53/0.5=106
2. Sqrt(106) = 10,295630140987000315797369464198
3. 4\*2.3 = 9.2
4. 10,295630140987000315797369464198 + (-11.1) = -0,80436985901299968420263053580245
5. -0,80436985901299968420263053580245 – 9.2 = -10,004369859012999684202630535802
6. 2.3\*5.6 =12.88
7. 1+12.88=13.88
8. -10,004369859012999684202630535802/13.88 = -0,72077592644185875687338836713271

Друга підстановка:

( sqrt(53/1.5) + (2.3) – 4\*3.3) / (1+ 3.3\*(-1.1))

1. 53/1.5=35,333333333333333333333333333333
2. Sqrt(35,333333333333333333333333333333) = 5,9441848333756692944167913893809
3. 4\*3.3=13.2
4. 5,9441848333756692944167913893809 + 2.3 = 8,2441848333756692944167913893809
5. 8,2441848333756692944167913893809 – 13.2 = -4,9558151666243307055832086106191
6. 3.3\*(-1.1) = -3.63
7. 1+-3.63 = -2.63
8. -4,9558151666243307055832086106191/(-2.63) = 1,8843403675377683291190907264711

Третя підстановка:

( sqrt(53/3.5) + (12.22) – 4\*5.1) / (1+ 5.1\*(2.2))

1. 53/3.5=15,142857142857142857142857142857
2. Sqrt(15,142857142857142857142857142857) = 3,8913824205360673928386982964774
3. 4\*5.1=20.4
4. 3,8913824205360673928386982964774 + 12.22 = 16,111382420536067392838698296477
5. 16,111382420536067392838698296477 – 20.4 = -4,288617579463932607161301703523
6. 5.1\*(2.2) = 11.22
7. 1+11.22= -12.22
8. -4,288617579463932607161301703523/12.22 = -0,35095070208379133904756969750597

Четверта підстановка:

( sqrt(53/(-4.3) + (-14.65) – 4\*1.5) / (1+ 1.5\*(-13.2))

1. 53/(-4.3)=- 12,325581395348837209302325581395

Підкорений вираз від’ємний подальший розв’язок не має сенсу.

П’ята підстановка:

( sqrt(53/(42.1) + (1.3) – 4\*5.5) / (1+ 5.5\*1.5)

1. 53/42.1=1,2589073634204275534441805225653
2. Sqrt(1,2589073634204275534441805225653) = 1,1220104114581234818124621652808
3. 4\*5.5=22
4. 1,1220104114581234818124621652808 + 1.3 = 2,4220104114581234818124621652808
5. 2,4220104114581234818124621652808 – 22 = -19,577989588541876518187537834719
6. 5.5\*(1.5) = 8.25
7. 1+7.75= 9.25
8. -19,577989588541876518187537834719/9.25 = -2,1165394149775003641283824686183

Проведемо округлення результатів до 16 знаків після коми:

1. -0.7207759264418588
2. 1.884340367537768
3. -0.3509507920837913
4. –
5. -2.1165394149775004

Лістинг програми

7-21-IP-03-Yanyshyn.asm

.686

.model flat, stdcall

option casemap: none

include \masm32\include\masm32rt.inc

public BValues, CValues, counter, Pointer

extern DenominatorProc:PROTO

.data?

GlobalBuf dq 1024 DUP(?)

BufFormat dq 1024 DUP(?)

BufForResult dq 1024 DUP(?)

SqrtBuf dt ?

SubBuf dt ?

DenominatorBuf dt ?

Pointer dd ?

counter dd ?

.data

AValues dq 0.5, 1.5, 3.5, -4.3, 42.1

BValues dq 2.3, 3.3, 5.1, 1.5, 5.5

CValues dq 5.6, -1.1, 2.2, -13.2, 1.5

DValues dq -11.1, 2.3, 12.22, -14.65, 1.3

ResultFormat db "Result: %s", 13, 13, 0

MyVariant db "Variant 21: ( sqrt(53/a) + d - 4\*b ) / ( 1+b\*c)", 13, 13, 0

MsgBoxCaption db "7-23-IP-03-Yanyshyn", 0

Error db "Error! Change variables!", 0

ResultValue dq 0,0

Four dq 4.0

FiftyThree dq 53.0

.code

SqrtProc proc

push ebp ;пролог

|  |  |
| --- | --- |
| max | max |
|  |  |
| Адрес AValues | Адрес початку масиву AValues |
| Адрес FiftyThree | Адрес по якому знаходиться число 53 |
| Адрес SqrtBuf | Буфер для збереження результату обчислення з процедури |
| EDI | Лічильник |
| EIP | Адреса повернення з процедури |
| EBP | Поточне значення EBP |
|  |  |
| min | min |

**;ESP -> min**

mov ebp, esp

mov esi, [ebp + 8] ;edi ->esi

mov ecx, [ebp + 16] ;53 -> ecx

mov ebx, [ebp + 20] ;A ->ebx

fld qword ptr [ecx] ;53 -> st(0)

fdiv qword ptr [ebx + esi \* 8] ;53/A -> st(0)

fsqrt ;sqrt(42/b) -> st(0)

mov ebx, [ebp + 12] ;SqrtBuf -> ebx

fstp TBYTE ptr [ebx] ;sqrt(42/b) -> SqrtBuf

pop ebp ;Епілог

ret 16

SqrtProc endp

|  |  |
| --- | --- |
| max | max |
|  |  |
| Адрес AValues | Адрес початку масиву AValues |
| Адрес FiftyThree | Адрес по якому знаходиться число 53 |
| Адрес SqrtBuf | Буфер для збереження результату обчислення з процедури |
| EDI | Лічильник |
| EIP | Адреса повернення з процедури |
| EBP | Поточне значення EBP |
|  |  |
| min | min |

**;ESP -> max**

SubProc proc

finit

fld qword ptr [ecx + edi\*8] ;D -> st(0)

fld qword ptr [eax + edi\*8] ;B -> st(0), D -> st(1)

fld qword ptr [edx] ;4 -> st(0), B -> st(1), D -> st(2)

fmul ;4\*B -> st(0), D -> st(1)

fsub

fstp TBYTE ptr [ebx]

ret

SubProc endp

start:

mov edi, 0

invoke szCatStr, offset GlobalBuf, offset MyVariant

Calculate:

finit

fld BValues[8\*edi] ;B -> st(0)

fld CValues[8\*edi] ;C -> st(0), B -> st(1)

fmul ;B \* C -> st(0)

fld1 ;1 -> st(0), B \* C -> st(1)

fadd ;1+B\*C -> st(0)

ftst

fstsw ax

sahf

je errorBlock

fld AValues[8\*edi] ;A -> st(0)

ftst

fstsw ax

sahf

jb errorBlock

finit

push offset AValues

push offset FiftyThree

push offset SqrtBuf

push edi

call SqrtProc

mov ecx, offset DValues

mov eax, offset BValues

mov ebx, offset SubBuf

mov edx, offset Four

call SubProc

push offset DenominatorBuf

pop Pointer

mov counter, edi

call DenominatorProc

finit

fld SqrtBuf

fld SubBuf

fadd

fld DenominatorBuf

fdiv

fstp ResultValue ;st(0) -> ResultValue

invoke FloatToStr2, ResultValue, offset BufForResult

invoke wsprintf, offset BufFormat , offset ResultFormat, offset BufForResult

jmp final

errorBlock:

invoke wsprintf, offset BufFormat , offset ResultFormat, offset Error

final:

invoke szCatStr, addr GlobalBuf, addr BufFormat

inc edi

.if edi<5

jmp Calculate

.endif

invoke MessageBox, 0, addr GlobalBuf, addr MsgBoxCaption, 0

invoke ExitProcess, 0

end start

7-21-IP-03-Yanyshyn2.asm

.686

.model flat, stdcall

option casemap: none

public DenominatorProc

extern BValues:QWORD, CValues:QWORD, Counter:DWORD, BufPointer:DWORD

.code

DenominatorProc proc

mov edi, Counter

mov eax, BufPointer

fld BValues[8\*edi] ;B -> st(0)

fld CValues[8\*edi] ;C -> st(0), B ->st(1)

fmul ;B\*C -> st(0)

fld1 ;1 -> st(0), B\*C -> st(1)

fadd ;B\*C+1 -> st(0)

fstp TBYTE PTR [eax]

ret

DenominatorProc endp

End

7-21-IP-03-Yanyshyn2.bat

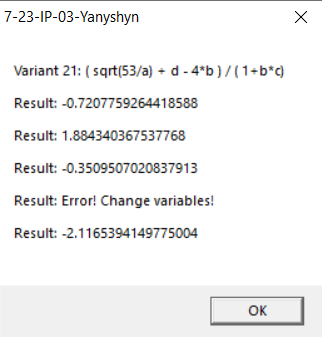
@echo off

\masm32\bin\ml /c /coff "7-21-IP-03-Yanyshyn.asm" "7-21-IP-03-Yanyshyn2.asm "

\masm32\bin\link /subsystem:windows "7-21-IP-03-Yanyshyn.obj" "7-21-IP-03-Yanyshyn2.obj"

PAUSE

Результати тестування



Висновок

На лабораторній роботі я вивчав прийоми модульного програмування, методи звернення до процедур і передачу в них параметрів. Результати контрольних розрахунків збігаються з результатами роботи програми, отже програма написана коректно.