Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение  
высшего образования   
«Алтайский государственный технический университет им. И. И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчет защищен с оценкой \_\_\_\_\_

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Отчет

По лабораторной работе №2

**«Разработка простых программ для DOS в среде ассемблера»**

по дисциплине «Машинно-зависимые языки программирования»

Студент группы ПИ-92 Гроссу Е.С.

Преподаватель доцент,к.т.н. Боровцов Е.Г.

Барнаул 2021

**Задание :**

Реализация простых программ, реализующих линейный, ветвящийся и циклический алгоритм с использованием прерываний DOS для организации ввода/вывода на экран.

**Вариант 7:**

Y=a-(b+c)2/d

**Код программы:**

.model tiny ; MASM's support for COM files

.dosseg ; Make sure you are using dos segment CODE, DATA + STACK

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Our data section. Here we declare our strings for our console message

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**.**data

A dw ?

B dw ?

C\_1 dw ?

D dw ?

Y dw ?

input\_buf db 06,00,5 dup(?)

in\_A db 0ah, 0dh, 'Input A :$'

in\_B db 0ah, 0dh, 'Input B :$'

in\_C db 0ah, 0dh, 'Input C :$'

in\_D db 0ah, 0dh, 'Input D :$'

answer db 7 dup(?), '$' ;ввод

messout db 0dh, 0ah, 'result : $' ;вывод

.code

.startup

beg: jmp start

include bin2str.asm

include str2bin.asm

; ==================ввод=====================================

start: lea dx,in\_A ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov A,ax ; запомнить число в переменной A

; ==================ввод=====================================

lea dx,in\_B ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov B,ax ; запомнить число в переменной B

; ==================ввод=====================================

lea dx,in\_C ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov C\_1,ax ; запомнить число в переменной C

; ==================ввод=====================================

lea dx,in\_D ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov D,ax ; запомнить число в переменной D

;Y=A-(B+C)^2/D

mov ax,B ; в регистр ax заносится B

mov bx,C\_1 ; в регистр bx заносится C

add ax,bx ;B+C

imul ax ;(B+C)^2

idiv D ;(B+C)^2/D

mov bx,A ; в регистр ax заносится A

sub bx,ax ;A-(B+C)^2/D

mov Y,bx ;результат поместить в Y

mov ax,bx ;результат в ax

lea bx,answer ; поместить в BX адрес буфера для символьного представления

call bin2str ; обратиться к подпрограмме преобразования

lea dx,messout ; в dx - адрес буфера вывода

mov ah,09h ; в ah - номер функции вывода на экран

int 21h ; обратиться к функции вывода через 21 прерывание

lea dx,answer+1

mov ah,09h

int 21h

int 20h ; а затем завершаем работу

end

**Набор исходных данных для положительного результата:**

|  |  |  |
| --- | --- | --- |
| значение | 10 с.с. | 16 с.с. |
| А | 5 | 05 |
| В | 1 | 01 |
| С | 1 | 01 |
| D | 2 | 02 |
| res | 3 | 03 |

**Набор исходных данных для отрицательного результата:**

|  |  |  |
| --- | --- | --- |
| значение | 10 с.с. | 16 с.с. |
| А | 0 | 00 |
| В | 1 | 01 |
| С | 1 | 01 |
| D | 1 | 01 |
| res | -4 | fc |

**Набор исходных данных для нулевого результата:**

|  |  |  |
| --- | --- | --- |
| значение | 10 с.с. | 16 с.с. |
| А | 3 | 03 |
| В | 2 | 02 |
| С | 1 | 01 |
| D | 3 | 02 |
| res | 0 | 00 |

**Тест программы:**

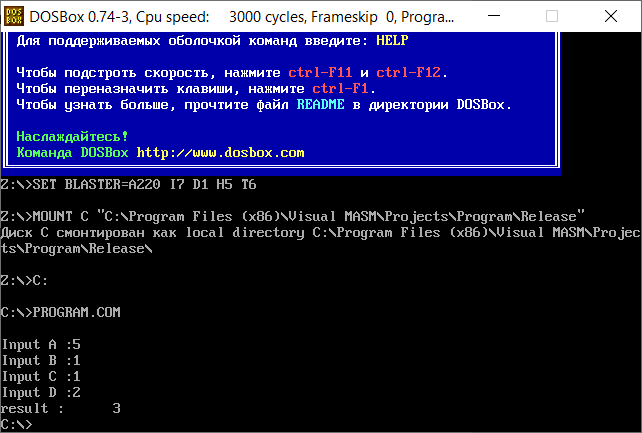


Рис1.1: положительный результат

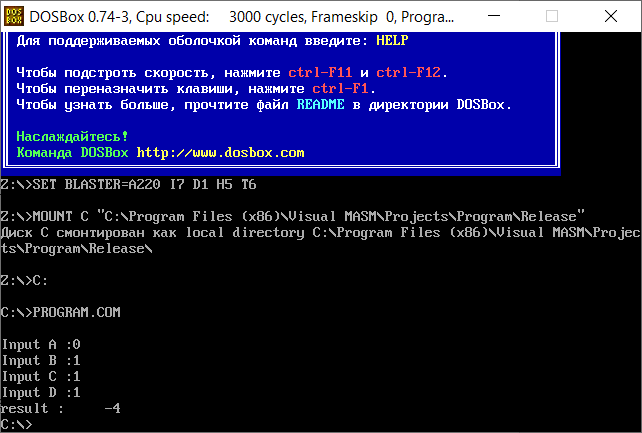


Рис1.2: отрицательный результат

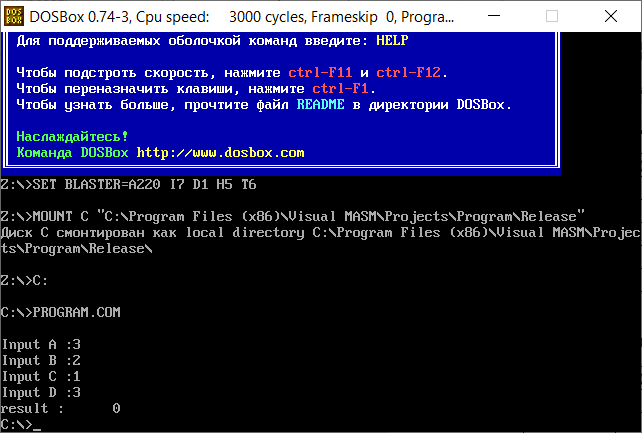
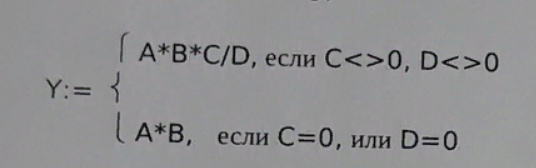


Рис1.3: нулевой результат

**Задание 2:** Условный оператор.

**Вариант 5:**



**Текст программы:**

.model tiny ; MASM's support for COM files

.dosseg ; Make sure you are using dos segment CODE, DATA + STACK

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Our data section. Here we declare our strings for our console message

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**.**data

A dw ?

B dw ?

C\_1 dw ?

D dw ?

Y dw ?

input\_buf db 06,00,5 dup(?)

in\_A db 0ah, 0dh, 'Input A :$'

in\_B db 0ah, 0dh, 'Input B :$'

in\_C db 0ah, 0dh, 'Input C :$'

in\_D db 0ah, 0dh, 'Input D :$'

answer db 7 dup(?), '$' ;ввод

messout db 0dh, 0ah, 'result : $' ;вывод

.code

.startup

beg: jmp start

include bin2str.asm

include str2bin.asm

; ==================ввод=====================================

start: lea dx,in\_A ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov A,ax ; запомнить число в переменной A

; ==================ввод=====================================

lea dx,in\_B ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov B,ax ; запомнить число в переменной B

; ==================ввод=====================================

lea dx,in\_C ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov C\_1,ax ; запомнить число в переменной C

; ==================ввод=====================================

lea dx,in\_D ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov D,ax ; запомнить число в переменной D

;if C=0 or D=0 A\*B

;If C!=0 and D!=0 A\*b\*c/D

mov ax,A ;загрузить в регистр ax A

imul B ;A\*B

mov bx,C\_1 ;загрузить в регистр bx C

cmp bx,0 ; Сравнить значение C c 0

jz output ;Если 0 то пререход по мектке

mov bx,D ;загрузить в регист bx D

cmp bx,0 ;сравнить згачение D c 0

jz output ;Если 0 то переход по метке

imul C\_1 ;A\*B\*C

idiv D ;A\*B\*C/D

output: mov Y,ax ; в Y занести результат

lea bx,answer ; поместить в BX адрес буфера для символьного представления

call bin2str ; обратиться к подпрограмме преобразования

lea dx,messout ; в dx - адрес буфера вывода

mov ah,09h ; в ah - номер функции вывода на экран

int 21h ; обратиться к функции вывода через 21 прерывание

lea dx,answer+1

mov ah,09h

int 21h

int 20h ; а затем завершаем работу

end

**Набор исходных данных для условия C<>0, D<>0**

|  |  |  |
| --- | --- | --- |
| значение | 10 с.с. | 16с.с |
| А | 1 | 01 |
| В | 2 | 02 |
| С | 3 | 03 |
| D | 3 | 03 |
| res | 2 | 02 |

**Набор исходных данных для условия C=0 или D=0**

|  |  |  |
| --- | --- | --- |
| значение | 10 с.с. | 16с.с |
| А | 2 | 02 |
| В | 3 | 03 |
| С | 0 | 00 |
| D | 4 | 04 |
| res | 6 | 06 |

**Тест программы:**

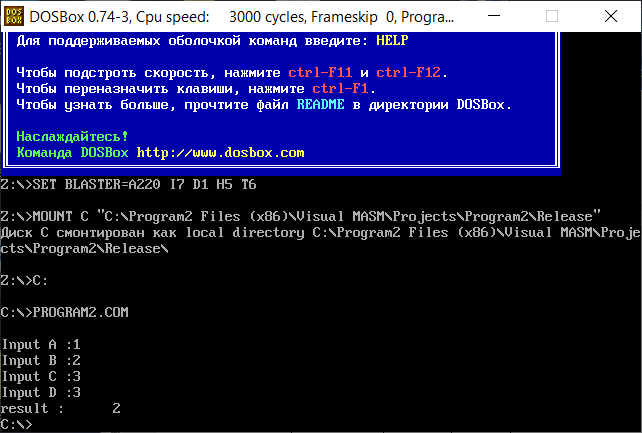


Рис2.1: Для условия C<>0, D<>0

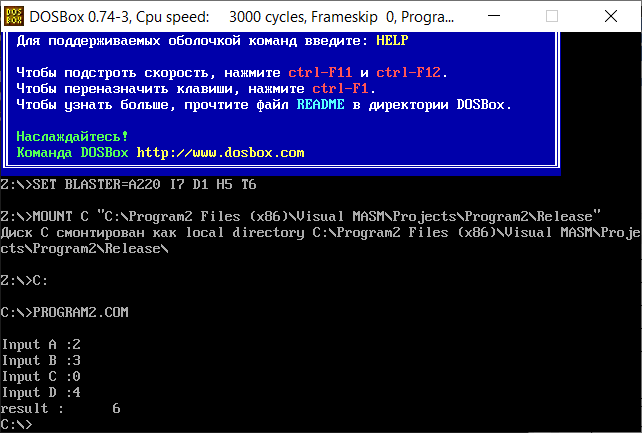
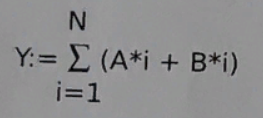


Рис2.2: Для условия С=0 или D=0

**Цикл: **

**Текст программы:**

.model tiny ; MASM's support for COM files

.dosseg ; Make sure you are using dos segment CODE, DATA + STACK

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Our data section. Here we declare our strings for our console message

; \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.data

N dw ?

A dw ?

B dw ?

Y dw ?

input\_buf db 06,00,5 dup(?)

in\_N db 0ah, 0dh, 'Input N :$'

in\_A db 0ah, 0dh, 'Input A :$'

in\_B db 0ah, 0dh, 'Input B :$'

answer db 7 dup(?), '$' ;ввод

messout db 0dh, 0ah, 'result : $' ;вывод

.code

.startup

beg: jmp start

include bin2str.asm

include str2bin.asm

; ==================ввод=====================================

start : lea dx,in\_N ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov N,ax ; запомнить число в переменной A

; ==================ввод=====================================

lea dx,in\_A ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov A,ax ; запомнить число в переменной B

; ==================ввод=====================================

lea dx,in\_B ; загрузка адреса буфера

mov ah,09h ; в AH - номер функции вывода строки

int 21h ; вызов прерывания DOS для вывода строки на экран

lea dx,input\_buf ; в dx - адрес буфера ввода

mov ah,0ah ; в AH - номер функции ввода числа с клавиатуры

int 21h ; вызов прерывания ввода числа с клавиатуры

;==========преобразование строки в число=======================

mov bx,dx ; перегрузить в bx адрес буфера

inc bx ; увеличить адрес на единицу

call str2bin ; обратиться к подпрограмме преобразования

mov B,ax ; запомнить число в переменной B

;Y=Sum i=1 for N (A+B)\*i

mov bx,0 ;обнулить регистр bx для результата

mov cx,N ;занести в регистр cx счетчик цикла

cmp cx,0 ;проверить N==0?

jz output ;если N=0 переход по метке

mov ax,A ;загрузить в регистр ax A

add ax,B ;A+B

mov A,ax ;загрузить сумму в ячейку A

cycl: push cx ;сохранить стек счетчик цикла

mov ax,A ; загрузить в регистр ax A

imul cx ;(A+B)\*i

add bx,ax ;Y+=(A+B)\*i

pop cx ;востановить счетчик цикла

loop cycl ;зациклить по метке

output:mov Y,bx ; поместить результат в Y

mov ax,bx ;результат в ax

lea bx,answer ; поместить в BX адрес буфера для символьного представления

call bin2str ; обратиться к подпрограмме преобразования

lea dx,messout ; в dx - адрес буфера вывода

mov ah,09h ; в ah - номер функции вывода на экран

int 21h ; обратиться к функции вывода через 21 прерывание

lea dx,answer+1

mov ah,09h

int 21h

int 20h ; а затем завершаем работу

end

**Набор исходных значений и результат для того чтобы цикл проработал >1 раза**

|  |  |  |
| --- | --- | --- |
| Значение | 10с.с. | 16с.с. |
| N | 2 | 02 |
| A | 0 | 00 |
| B | 3 | 03 |
| Y | 9 | 09 |

**Набор исходных значений и результат для того чтобы цикл проработал 1 раз**

|  |  |  |
| --- | --- | --- |
| Значение | 10с.с. | 16с.с. |
| N | 1 | 01 |
| A | 1 | 01 |
| B | 2 | 02 |
| Y | 3 | 03 |

**Набор исходных значений и результат для того чтобы цикл проработал 0 раз**

|  |  |  |
| --- | --- | --- |
| Значение | 10с.с. | 16с.с. |
| N | 0 | 00 |
| A | 4 | 04 |
| B | 5 | 05 |
| Y | 0 | 00 |

**Тест программы:**

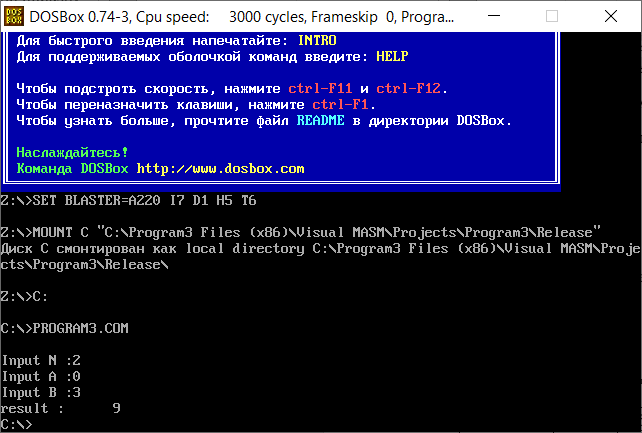


Рис3.1: Работа цикла больше 1 раза

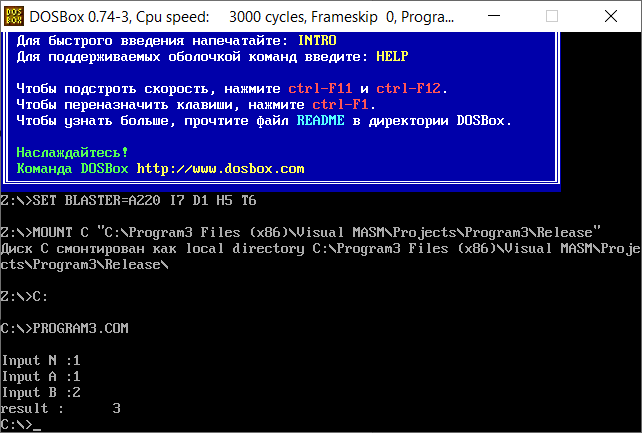


Рис3.2: Работа цикла 1 раз

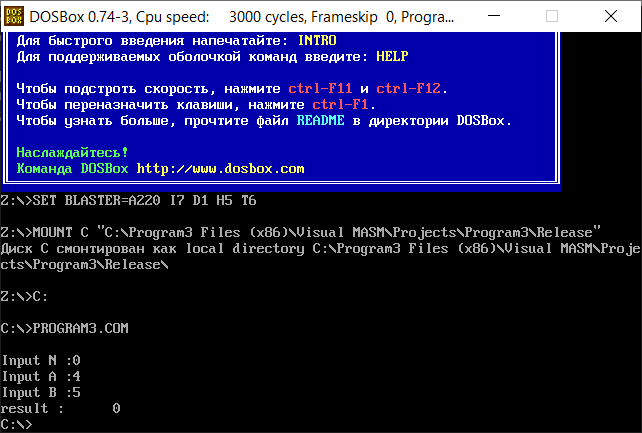


Рис3.3: Работа цикла 0 раз

Юров Виктор ИвановичAssembler. Учебник для вузов2 издание

Зубков, С. В. Assembler. Для DOS, Windows и Unix : учебное пособие / С. В. Зубков. — Москва : ДМК Пресс, 2008. — 640 с. — ISBN 5-94074-259-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/1243 (дата обращения: 18.12.2020). — Режим доступа: для авториз. Пользователей.

от польз. Eugeny польз. все: 5:29 PM

Аблязов, Р. З. Программирование на ассемблере на платформе x86-64 / Р. З. Аблязов. — Москва : ДМК Пресс, 2011. — 304 с. — ISBN 978-5-94074-676-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/1273 (дата обращения: 18.12.2020). — Режим доступа: для авториз. Пользователей.

от польз. Eugeny польз. все: 5:30 PM

Пильщиков, В.Н. Программирование на языке ассемблера IBM PC : учебное пособие / В.Н. Пильщиков. – Москва : Диалог-МИФИ, 2014. – 288 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=447687 (дата обращения: 31.01.2021). – Библиогр. в кн. – ISBN 5-86404-051-7. – Текст : электронный.