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 №4

# «Организация подпрограммы и механизма передачи параметров»

по дисциплине «Архитектура ЭВМ – 1 семестр»

Студент группы ПИ-02 Чередов Р.А.

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

Барнаул 2021

# Задание:

Для заданного выражения реализовать программу с использованием подпрограмм тремя вариантами, используя различные механизмы передачи параметров.

https://sun9-15.userapi.com/impf/gaJ4NG26Z0BQTjx-ndXRdkSiUUVx9KhdrGnHiQ/3p0SWcRI0mg.jpg?size=593x27&quality=96&sign=7f7b8959056aa56a89d20f5146ea61fb&type=album

# Передача параметров через общую область памяти

# 1 00 40 in 0 ; основание N

# 2 01 18 34 stor a, N; запись основания в общую область N

# 3 03 41 in 1 ; основание X

# 4 04 18 32 stor a, X ; запись основания в общую область X

# 5 06 42 in 2 ;основание Y

# 6 07 18 33 stor a, Y ;запись основания в общую область Y

# 8 09 10 33 load a, Y ;Y в ra

# 9 0b 18 36 stor a, A ;запись основания в A

# 10 0d 11 32 load b, X ;X в rb

# 11 0f 81 mur a, b; rb\*rb(x^2)

# 12 10 19 37 stor b, Z; сохранить rb в Z

# 13 12 c0 4b call power; вызов power

# 14 14 11 38 load b, REZ; загрузить rez

# 15 16 19 39 stor b, F; сохранить в F

# 16 18 11 32 load b, X; загрузить X

# 17 1a 79 01 sbi b, 1; x-1

# 18 1c 19 37 stor b, Z; сохранить rb в Z

# 19 1e c0 4b call power; вызов power

# 20 20 11 39 load b, F; загрузить F

# 21 22 71 38 sbm b, REZ; f-rez

# 22 24 19 39 stor b, F; сохранить в F

# 23 26 11 32 load b,; X загрузить X

# 24 28 19 37 stor b, Z; сохранить rb в Z

# 25 2a c0 4b call power; вызов power

# 26 2c 10 39 load a, F; загрузить F

# 27 2e 58 38 adm a, REZ; f+rez

# 28 30 4f out 7

# 29 31 f8 stop

# 30 32 00 X: .ds 1

# 31 33 00 Y: .ds 1

# 32 34 00 N: .ds 1

# 33 35 00 B: .ds 1

# 34 36 00 A: .ds 1

# 35 37 00 Z: .ds 1

# 36 38 00 REZ: .ds 1

# 37 39 00 F: .ds 1

# 39 3a 20 step: push a ; ra в стек

# 40 3b 21 push b; rb в стек

# 41 3c 10 34 load a, N;

# 42 3e 31 01 mvi b, 1;

# 43 40 20 cycl1: push a; ra в стек

# 44 41 88 35 mum a, B; умножить rb\*B

# 45 43 28 pop a; восстановить из стека содержимое регистра A

# 46 44 e0 40 loop a, cycl1; начало цикла

# 47 46 19 38 stor b, REZ; сохранить в REZ

# 48 48 29 pop b; восстановить из стека содержимое регистра B

# 49 49 28 pop a; восстановить из стека содержимое регистра A

# 50 4a c8 ret; возврат из подпрограммы

# 52 4b 20 power: push a; сохранить содержимое регистра A в стеке

# 53 4c 21 push b; сохранить содержимое регистра В в стеке

# 54 4d 10 37 load a, Z; загрузить в ra Z

# 55 4f 18 35 stor a, B; сохранить B

# 56 51 c0 3a call step; вызов step

# 57 53 11 38 load b, REZ; загрузить в rb REZ

# 58 55 10 36 load a, A; загрузить в ra A

# 59 57 18 35 stor a, B; сохранить в В

# 60 59 80 mur a, a; ra\*rb

# 61 5a c0 3a call step; вызов step

# 62 5c 10 38 load a, REZ; загрузить rez в ra

# 63 5e 6a sbr b, a; ra-rb

# 64 5f 61 01 adi b, 1; rb +1

# 65 61 19 38 stor b, REZ; сохранить в REZ

# 66 63 29 pop b; восстановить из стека содержимое регистра B

# 67 64 28 pop a; восстановить из стека содержимое регистра A

# 68 65 c8 ret; возврат из подпрограммы

# 

# Y=p(2,1^2)-p(2,0)+(2,1)=2\*1^2-2^2+1-0+2^2-1+2-2^2+1=1

# 

# 

# Y=p(3,2^2)-p(3,1)+(3,2)=3\*(2^2)-3+1-3+3-1+3\*2-3+1=13(0d)

# Передача параметров через регистры

# 1 00 40 in 0; основание N

# 2 01 18 28 stor a, N; запись основания в общую область N

# 3 03 41 in 1; основание X

# 4 04 18 26 stor a, X; запись основания в общую область X

# 5 06 42 in 2; основание Y

# 6 07 18 27 stor a, Y; запись основания в общую область Y

# 8 09 11 26 load b, X; X в rb

# 9 0b 81 mur a, b; x^2

# 10 0c 10 27 load a, Y; Y в ra

# 11 0e c0 35 call power; вызов power

# 12 10 20 push a; ra в стек

# 13 11 11 26 load b, X; X в rb

# 14 13 79 01 sbi b, 1; x-1

# 15 15 10 27 load a, Y; Y в ra

# 16 17 c0 35 call power; вызов power

# 17 19 29 pop b; восстановить из стека содержимое регистра B

# 18 1a 6a sbr b, a; ra-rb

# 19 1b 21 push b; rb в стек

# 20 1c 11 26 load b, X; X в rb

# 21 1e 10 27 load a, Y; Y в ra

# 22 20 c0 35 call power; вызов power

# 23 22 29 pop b; ; восстановить из стека содержимое регистра B

# 24 23 51 adr a, b; ra+rb

# 25 24 4f out 7;

# 26 25 f8 stop;

# 27 26 00 X: .ds 1;

# 28 27 00 Y: .ds 1;

# 29 28 00 N: .ds 1;

# 31 29 38 step: xchg; ra⬄rb

# 32 2a 21 push b; rb в стек

# 33 2b 31 01 mvi b, 1; rb = 1

# 34 2d 20 cycl1: push a; ra в стек

# 35 2e 8c 82 mum a, 2(sp) ; rb\* 2 элемент в стеке

# 36 30 28 pop a; восстановить из стека содержимое регистра a

# 37 31 e0 2d loop a, cycl1; начало цикла

# 38 33 28 pop a; восстановить из стека содержимое регистра a

# 39 34 c8 ret; возврат из подпрограммы

# 41 35 20 power: push a; ra в стек

# 42 36 21 push b; rb в стек

# 43 37 09 mov a, b; ra=rb

# 44 38 11 28 load b, N; N в rb

# 45 3a c0 29 call step; вызов step

# 46 3c 8c 82 mum a, 2(sp) ;

# 47 3e 28 pop a; восстановить из стека содержимое регистра a

# 48 3f 28 pop a; восстановить из стека содержимое регистра a

# 49 40 21 push b ;rb в стек

# 50 41 11 28 load b, N; N в rb

# 51 43 c0 29 call step; вызов step

# 52 45 28 pop a; восстановить из стека содержимое регистра a

# 53 46 69 sbr a, b; ra-rb

# 54 47 60 01 adi a, 1; ra+1

# 55 49 c8 ret; возврат из подпрограммы

# 

# Y=p(3,2^2)-p(3,1)+(3,2)=3\*(2^2)-3+1-3+3-1+3\*2-3+1=13(0d)

# 

# Y=p(2,1^2)-p(2,0)+(2,1)=2\*1^2-2^2+1-0+2^2-1+2-2^2+1=1

# Передача параметров через стек

# 1 00 20 push a; rA в стек

# 2 01 40 in 0; основание N

# 3 02 18 42 stor a, N; запись основания в общую область N

# 4 04 41 in 1; основание X

# 5 05 18 40 stor a, X; запись основания в общую область X

# 6 07 42 in 2; основание Y

# 7 08 18 41 stor a, Y; запись основания в общую область Y

# 8 0a 11 40 load b, X; X в rb

# 9 0c 81 mur a, b;x^2

# 10 0d 10 41 load a, Y; Y в ra

# 11 0f 20 push a; rA в стек

# 12 10 21 push b; rb в стек

# 13 11 c0 56 call power; вызов power

# 14 13 29 pop b; восстановить из стека содержимое регистра b

# 15 14 29 pop b; восстановить из стека содержимое регистра b

# 16 15 28 pop a; восстановить из стека содержимое регистра a

# 17 16 18 43 stor a, fst; запись основания в общую область fst

# 18 18 30 00 mvi a, 0; rA=0

# 19 1a 20 push a; rA в стек

# 20 1b 10 41 load a, Y; Y в ra

# 21 1d 11 40 load b, X; X в rb

# 22 1f 79 01 sbi b, 1;

# 23 21 20 push a; rA в стек

# 24 22 21 push b; rb в стек

# 25 23 c0 56 call power; вызов power

# 26 25 29 pop b; восстановить из стека содержимое регистра b

# 27 26 29 pop b; восстановить из стека содержимое регистра b

# 28 27 29 pop b; восстановить из стека содержимое регистра b

# 29 28 10 43 load a, fst; загрузить переменную для сохранения результата

# 30 2a 69 sbr a, b; ra-rb

# 31 2b 18 43 stor a, fst; сохранить в переменную для результата

# 32 2d 30 00 mvi a, 0; обнуляем ra

# 33 2f 20 push a; rA в стек

# 34 30 10 41 load a, Y; загрузить в ra Y

# 35 32 11 40 load b, X; загрузить в ra X

# 36 34 20 push a; rA в стек

# 37 35 21 push b; rb в стек

# 38 36 c0 56 call power; вызов power

# 39 38 28 pop a; восстановить из стека содержимое регистра a

# 40 39 28 pop a; восстановить из стека содержимое регистра a

# 41 3a 28 pop a; восстановить из стека содержимое регистра a

# 42 3b 11 43 load b, fst; в rb вносим переменную для результата

# 43 3d 51 adr a, b;ra+rb

# 44 3e 4f out 7;

# 45 3f f8 stop;

# 46 40 00 X: .ds 1

# 47 41 00 Y: .ds 1

# 48 42 00 N: .ds 1

# 49 43 00 fst: .ds 1

# 51 44 15 82 step: load b, 2(sp);

# 52 46 14 83 load a, 3(sp);

# 53 48 78 01 sbi a, 1; ra-1

# 54 4a 19 55 stor b, mn; сохранить в mn

# 55 4c 20 cycl1: push a;

# 56 4d 88 55 mum a, mn;

# 57 4f 28 pop a; восстановить из стека содержимое регистра a

# 58 50 e0 4c loop a, cycl1;

# 59 52 1d 84 stor b, 4(sp);

# 60 54 c8 ret; возврат из подпрограммы

# 61 55 00 mn: .ds 1;

# 63 56 15 82 power: load b, 2(sp);

# 64 58 30 00 mvi a, 0; обнуляем

# 65 5a 20 push a; rA в стек

# 66 5b 10 42 load a, N; загружение N в ra

# 67 5d 20 push a; rA в стек

# 68 5e 21 push b; rb в стек

# 69 5f c0 44 call step; вызов step

# 70 61 28 pop a; восстановить из стека содержимое регистра a

# 71 62 28 pop a; восстановить из стека содержимое регистра a

# 72 63 29 pop b; восстановить из стека содержимое регистра b

# 73 64 8c 83 mum a, 3(sp);

# 74 66 19 7e stor b, sec;сохранить в sec

# 75 68 15 83 load b, 3(sp);

# 76 6a 30 00 mvi a, 0;обнуляем

# 77 6c 20 push a; rA в стек

# 78 6d 10 42 load a, N;N в ra

# 79 6f 20 push a; rA в стек

# 80 70 21 push b; rb в стек

# 81 71 c0 44 call step; вызов step

# 82 73 28 pop a; восстановить из стека содержимое регистра a

# 83 74 28 pop a; восстановить из стека содержимое регистра a

# 84 75 29 pop b; восстановить из стека содержимое регистра b

# 85 76 10 7e load a, sec; загрузить step

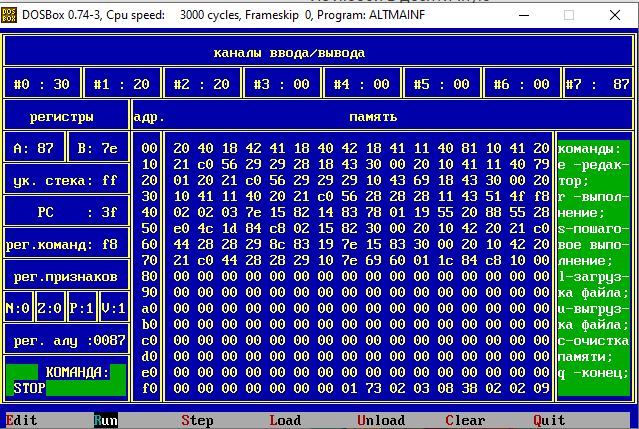
# 86 78 69 sbr a, b; ra-rb

# 87 79 60 01 adi a, 1;ra-1

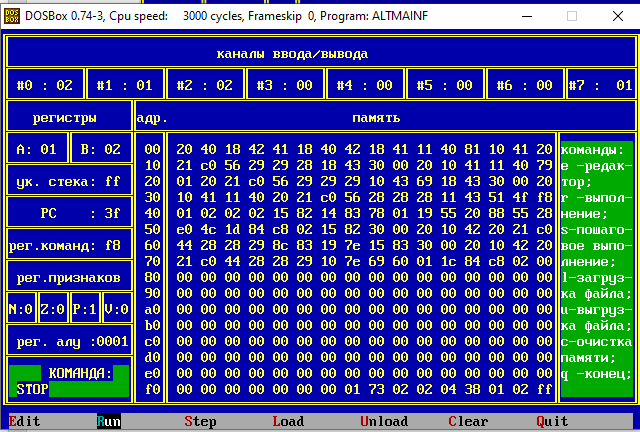
# 88 7b 1c 84 stor a, 4(sp);

# 89 7d c8 ret; возврат из подпрограммы

# 90 7e 00 sec: .ds 1



Y=(2,2^2)-(2,1)+(2,2)=2\*4^3-2^3+1-2+2^3-1+2\*2^3+1=135(87)



Y=p(2,1^2)-p(2,0)+(2,1)=2\*1^2-2^2+1-0+2^2-1+2-2^2+1=1