# Приложение А.

Программа на микроассемблере

;Написал Новотарский Кирилл, ИВ-73

;Функция Х=(В2 +A + sqrt(B))+sqrt(А)+(А-B)2

$mod51

;адреса рабочих регистров, через которые идет счет

Reg1 equ 40h

Reg2 equ 41h

Reg3 equ 42h

RegL1 equ 43h

RegL2 equ 44h

RegL3 equ 45h

RegR1 equ 4Bh

RegR2 equ 4Ch

RegR3 equ 4Dh

; значения А и B

ValueAdressA1 equ #00h

ValueAdressA2 equ #04h

ValueAdressB1 equ #00h

ValueAdressB2 equ #04h

ValuePoryadokB equ #01h

; адрес числа А

ADRESSA1 equ 0Ah

ADRESSA2 equ 0Bh

; адрес числа B

ADRESSB1 equ 0Ch

ADRESSB2 equ 0Dh

PB equ 0Eh

; адреса промежуточных результатов вычислений

ADRESSAB1 equ 0A0h

ADRESSAB2 equ 0A1h

ADRESSAB3 equ 0A2h

ADRESSBBASQB1 equ 0A3h

ADRESSBBASQB2 equ 0A4h

ADRESSBBASQB3 equ 0A5h

MSQABA1 equ 0A8h

MSQABA2 equ 0A9h

MSQABA3 equ 0AAh

; адрес окончательного результата

ADRESSRESULT1 equ 48h

ADRESSRESULT2 equ 49h

PODYADOKRESULT equ 4Ah

;СОБСТВЕННО ПРОГРАММА

;Загружаем исходные данные, описанные в константах

Mov ADRESSA1,VADRESSA1

Mov ADRESSA2,VADRESSA2

Mov ADRESSB1,VADRESSB1

Mov ADRESSB2,VADRESSB2

Mov PB, ValuePoryadokB

; ----------- Получаем B^2

Mov Reg1, ValueAdressB1 ; формування двох множників для процедури добутку

Mov Reg2, ValueAdressB2 ;

mov Reg3, #0h

Mov RegL1, ValueAdressB1 ;

Mov RegL2, ValueAdressB2 ;

Mov RegL3, VPb ;

Lcall multiply

mov ADRESSAB1,RegR1

mov ADRESSAB2,RegR2

mov ADRESSAB3,RegR3

;---------Получаем sqrt(А)----

mov Reg1, ValueAdressА1

mov Reg2, ValueAdressА2

mov Reg3, ValuePoryadokB

lcall sqrt

; ------Получаем B^2 + sqrt(A)

mov RegL1,RegR1

mov RegL2,RegR2

mov RegL3,RegR3

mov Reg1, ADRESSAB1

mov Reg2, ADRESSAB2

mov Reg3, ADRESSAB2

Lcall Add

mov ADRESSBBASQB1,RegR1

mov ADRESSBBASQB2,RegR2

mov ADRESSBBASQB3,RegR3

; ------Получаем B^2 + sqrt(A) + A

mov RegL1, ValueAdressА1

mov RegL2, ValueAdressА2

mov RegL3, ValuePoryadokB

mov Reg1, ADRESSAB1

mov Reg2, ADRESSAB2

mov Reg3, ADRESSAB2

Lcall Add

;---------Получаем sqrt(B)----

mov Reg1, ValueAdressB1

mov Reg2, ValueAdressB2

mov Reg3, ValuePoryadokB

lcall sqrt

; ------Получаем B^2 + sqrt(A) + A + sqrt(B)

mov RegL1,RegR1

mov RegL2,RegR2

mov RegL3,RegR3

mov Reg1, ADRESSAB1

mov Reg2, ADRESSAB2

mov Reg3, ADRESSAB2

Lcall Add

; -------Получаем A-B

mov RegL1, ValueAdressA1

mov RegL2, ValueAdressA2

mov RegL3, ValuePoryadokB

mov Reg1, ValueAdressB1

mov Reg2, ValueAdressB2

mov Reg3, ValuePoryadokB

Lcall Sub

; Получаем (A-B)^2 = (A-B)\*(A-B)

Mov Reg1, ADRESSAB1

Mov Reg2, ADRESSAB2

mov Reg3, #0h

Mov RegL1, ADRESSAB2

Mov RegL2, ADRESSAB1

Mov RegL3, ValuePoryadokB ;

Lcall multiply

; Окончательно суммируем для получения результата.

; Результат: X = B^2 + A + sqrt(B)+ sqrt(A) + (A-B)^2

mov RegL1,RegR1

mov RegL2,RegR2

mov RegL3,RegR3

mov Reg1, ADRESSAB1

mov Reg2, ADRESSAB2

mov Reg3, ADRESSAB2

Lcall Add

;Заносим результат в адреса для результата

mov ADRESSRESULT1,RegR1

mov ADRESSRESULT2,RegR2

mov PODYADOKRESULT,RegR3

;Прыгаем на выход

ljmp exit1

;------------------ ВСПОМОГАТЕЛЬНЫЕ ПРОЦЕДУРЫ

;Процедура вычитания

;Операнды - (Reg1-Reg3):A (RegL1-RegL3):B

;Результат в (RegR1-RegR3):Res

;формирование значения разницы порядков для формирования счетчика

sub:

mov a,Reg3

add a,RegL3

mov Reg3,a

jz addex

jnc add1

;в зависимости от значения разницы порядков числа А сдвигается либо влево, либо ;вправо

sub2: clr c

mov a,Reg1

rrc A

mov Reg1,a

mov A,Reg2

rrc A

mov Reg2,a

mov a,Reg3

inc a

mov Reg3,a

jnz add2

ljmp addex

sub1: clr c

mov a,Reg2

rlc A

mov Reg2,a

mov A,Reg1

rlc A

mov Reg1,a

mov a,Reg3

dec a

mov Reg3,a

jnz add1

;собственно сложения

subex: clr c

mov a,Reg2

addc a,RegL2

mov RegR2,a

mov a,Reg1

addc a,RegL1

mov RegR1,a

;Формирование порядка

mov a,RegL3

mov RegR3,a

ret

; Конец процедуры

;Процедура сложения

;Операнды - (Reg1-Reg3):A (RegL1-RegL3):B

;Результат в (RegR1-RegR3):Res

;формирование значения разницы порядков для формирования счетчика

add:

mov a,Reg3

subb a,RegL3

mov Reg3,a

jz addex

jnc add1

;в зависимости от значения разницы порядков числа А сдвигается либо влево, либо ;вправо

add2: clr c

mov a,Reg1

rrc A

mov Reg1,a

mov A,Reg2

rrc A

mov Reg2,a

mov a,Reg3

inc a

mov Reg3,a

jnz add2

ljmp addex

add1: clr c

mov a,Reg2

rlc A

mov Reg2,a

mov A,Reg1

rlc A

mov Reg1,a

mov a,Reg3

dec a

mov Reg3,a

jnz add1

;собственно сложения

addex: clr c

mov a,Reg2

addc a,RegL2

mov RegR2,a

mov a,Reg1

addc a,RegL1

mov RegR1,a

;Формирование порядка

mov a,RegL3

mov RegR3,a

ret

; Конец процедуры

;процедура извлечения корня

;Операнд (Reg1-Reg3):A , а результат (RegR1-RegR3):Res

sqrt:

;счетчик

mov @20h,#10h

;началньео значения -1

mov RegL2,#80h

;нормализация порядка

clr c

mov a,Reg3

mov c,acc.0

jnc sqvir

jmp viren:

sqvir:

inc a

mov Reg3,a

;зсув вправо

clr c

mov a,Reg1

rrc a

mov Reg1,a

mov a,Reg2

rrc a

mov Reg2,a

viren:

sqrt1:

;сдвиг влево

clr c

mov a,Reg2

rlc a

mov Reg2,a

mov a,Reg1

rlc a

mov Reg1,a

;сдвиг влево регистра ADRESSB

clr c

mov a,RegL3

rrc a

mov RegL3,a

mov a,@46h

rrc a

mov @46h,a

;Формирование след значения

mov a,RegL2

add a,Reg2

subb a,RegR2

subb a,RegR2

add a,@46h

mov RegL2,a

mov a,RegL1

add a,Reg1

subb a,RegR1

subb a,RegR1

add a,RegL3

mov RegL1,a

;Проверка значения

mov c,acc.7

jc res

;сдвиг влево, если дельта равно 0

clr c

mov a,RegR2

rlc a

mov RegR2,a

mov a,RegR1

rlc a

mov RegR1,a

mov a,RegL2

mov Reg2,a

mov a,RegL1

mov Reg1,a

ljmp next

res:

;сдви влево, если дельта = 1

mov a,RegR2

rlc a

mov RegR2,a

mov a,RegR1

rlc a

mov RegR1,a

;проверяем счетчик

next:

mov a,@20h

dec a

mov @20h,a

jnz sqrt1

ensqrt:

;формирование порядка результата

mov a,Reg3

rrc a

mov RegR3,a

ret

; Конец процедуры

; Процедура умножения

;Операнды (Reg1-Reg3):A (RegL1-RegL3):B

;Результат в: (RegR1-RegR3):Res

Multiply:

mov 19h,#10h; счетчким

;начало цикла умноежния

cycle1:

xch A,Reg2

mov c,acc.0

xch A,Reg2

;проверка значения разряда множимого

jc cycle2

add a,#0h

;сдвиг множимого вправо

xch A,Reg1

rrc A

xch A,Reg1

xch A,Reg2

rrc A

xch A,Reg2

;влево

add a,#0h

xch A,RegL2

rlc A

xch A,RegL2

xch A,RegL1

rlc A

xch A,RegL1

;прыгаем на проверку счетчика

ljmp mul1

cycle2:

;суммируем результат и множитель

add a,#0h

mov A,RegL2

add A,RegR2

mov RegR2,a

mov A,RegL1

addc A,RegR1

mov RegR1,a

add a,#0h

;сдвигаем множенное вправо

xch A,Reg1

rrc A

xch A,Reg1

xch A,Reg2

rrc A

xch A,Reg2

add a,#0h

;сдвиг множителя влево

xch A,RegL2

rlc A

xch A,RegL2

xch A,RegL1

rlc A

xch A,RegL1

;проверяем значение счетчика

mul1:

xch a,@19h

dec A

jz mul2

xch a,@19h

ljmp cycle1

mul2:

; равняем порядки

mov a,RegL3

add a,Reg3

mov RegR3,a

ret

;Конец процедуры

exit1:

end