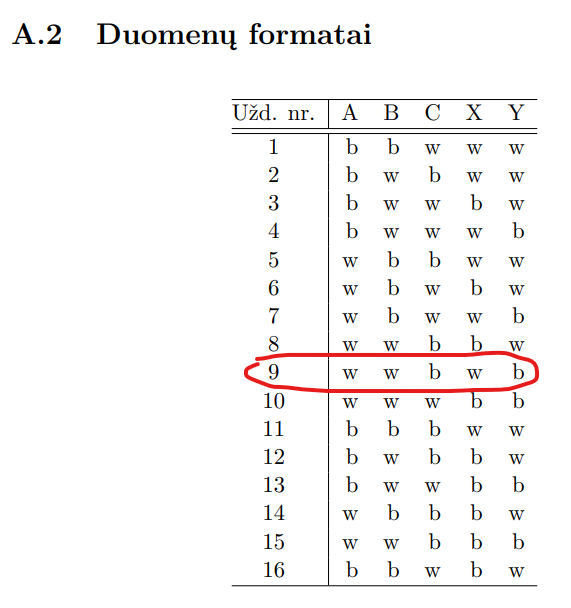
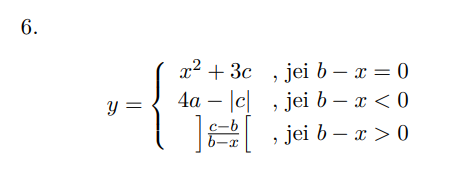
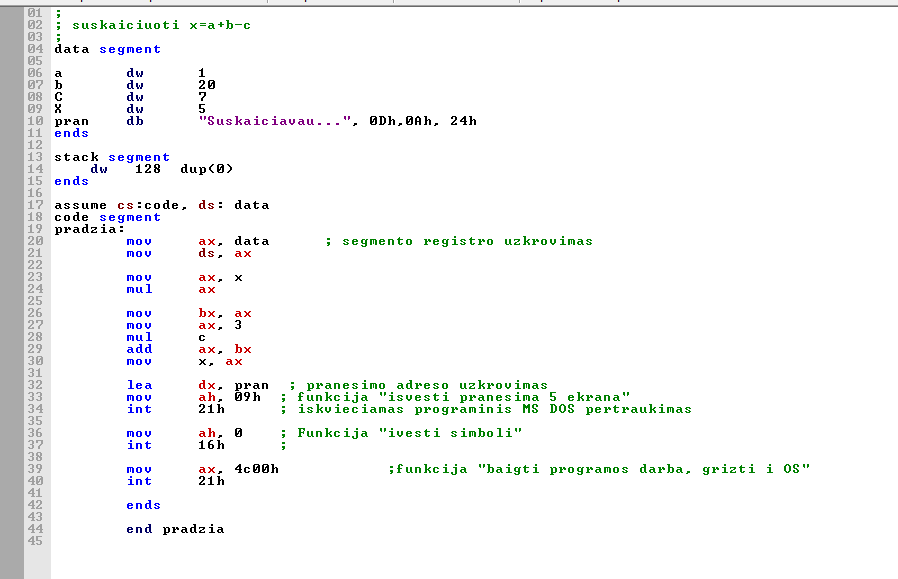
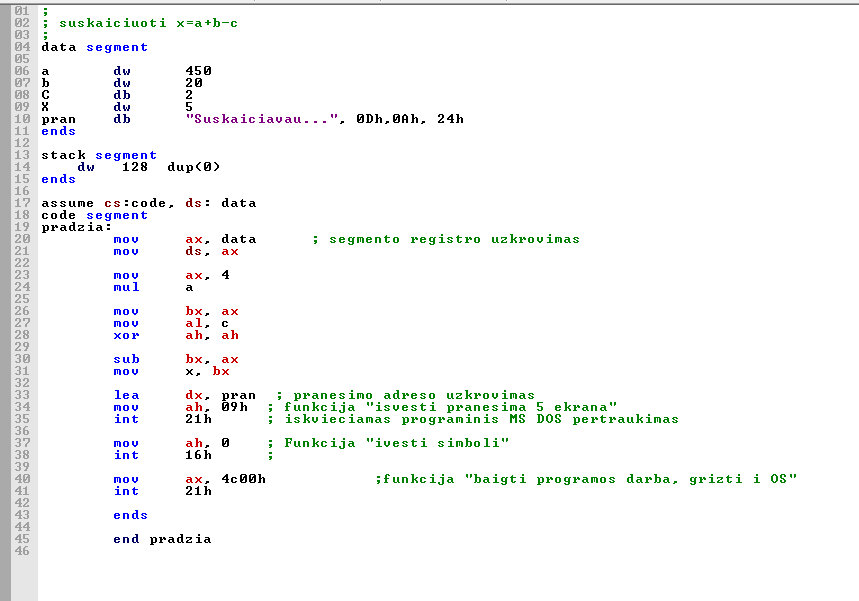
|216 | Liaudanskis Nedas IFF-1/9 | 6 | 9 | Be ženklo |









;

; suskaiciuoti / x^2+3\*c, kai b-x=0

; y = | 4\*a-|c|, kai b-x<0

; \ (c-b)/(b-x), kai b-x>0

; skaiciai be zenklo

; Duomenys a - w, b - w, c - b, x - w, y - b

stekas SEGMENT STACK

DB 256 DUP(0)

stekas ENDS

duom SEGMENT

a DW 1000 ;10000 ; perpildymo situacijai

b DW 1000

c Db 11

x DW 1000,2000,200,4100,1200,900,4500,600

kiek = ($-x)/2

y DB kiek dup(0AAh)

isvb DB 'x=',6 dup (?), ' y=',6 dup (?), 0Dh, 0Ah, '$'

perp DB 'Perpildymas', 0Dh, 0Ah, '$'

daln DB 'Dalyba is nulio', 0Dh, 0Ah, '$'

netb DB 'Netelpa i baita', 0Dh, 0Ah, '$'

spausk DB 'Skaiciavimas baigtas, spausk bet kuri klavisa,', 0Dh, 0Ah, '$'

duom ENDS

prog SEGMENT

assume ss:stekas, ds:duom, cs:prog

pr: MOV ax, duom

MOV ds, ax

XOR si, si ; (suma mod 2) si = 0

XOR di, di ; di = 0

c\_pr: MOV cx, kiek

JCXZ pab

;//////////////////////

cikl:

MOV ax, b

mov bx, x[si]

sub ax, bx

mov bx, 0

CMP bx,ax ;patikrina salyga b-x=0 NEVEIKIA CIA

JE f2

Jl f1

jge f3

;jeigu daugiau ///////////////////////////

;c-b

f1:

MOV al, c

mov ah, 0 ;????????????????????????????????????

sub ax, b ;c-b

JC kl1

; b-x

XCHG bx, ax

MOV ax, b

sub ax, x[si] ; b-x

JC kl1

; (c-b)/(b-x)

XCHG ax, bx

CMP bx, 0

;JE k12

xor dx, dx

div bx ; (c-b)/(b-x)

JC kl1

JMP re

;jeigu lygu////////////////////////////////

f2:

MOV ax, x[si]

MUL ax

JC kl1 ; sandauga netilpo i ax

mov bx, ax

mov ax, 3

mul c

JC kl1 ; sandauga netilpo i ax

add ax, bx

JC kl1

JMP re

;jeigu maziau//////////////////////////////

f3:

MOV ax, 4

MUL a

JC kl1 ; sandauga netilpo i ax

mov bx, ax;

mov al, c

xor ah, ah

sub bx, ax

JC kl1

mov ax, bx

re:

CMP ah, 0 ;ar telpa rezultatasi baita

JE ger

JMP kl3

ger: MOV y[di], al

INC si

INC si

INC di

LOOP cikl

pab:

;rezultatu isvedimas i ekrana

;============================

XOR si, si

XOR di, di

MOV cx, kiek

JCXZ is\_pab

is\_cikl:

MOV ax, x[si] ; isvedamas skaicius x yra ax reg.

PUSH ax

MOV bx, offset isvb+2

PUSH bx

CALL binasc

MOV al, y[di]

XOR ah, ah ; isvedamas skaicius y yra ax reg.

PUSH ax

MOV bx, offset isvb+11

PUSH bx

CALL binasc

MOV dx, offset isvb

MOV ah, 9h

INT 21h

;============================

INC si

INC si

INC di

LOOP is\_cikl

is\_pab:

;===== PAUZE ===================

;===== paspausti bet kuri klavisa ===

LEA dx, spausk

MOV ah, 9

INT 21h

MOV ah, 0

INT 16h

;============================

MOV ah, 4Ch ; programos pabaiga, grizti i OS

INT 21h

;============================

kl1: LEA dx, perp

MOV ah, 9

INT 21h

XOR al, al

JMP ger

kl2: LEA dx, daln

MOV ah, 9

INT 21h

XOR al, al

JMP ger

kl3: LEA dx, netb

MOV ah, 9

INT 21h

XOR al, al

JMP ger

; skaiciu vercia i desimtaine sist. ir issaugo

; ASCII kode. Parametrai perduodami per steka

; Pirmasis parametras ([bp+6])- verciamas skaicius

; Antrasis parametras ([bp+4])- vieta rezultatui

binasc PROC NEAR

PUSH bp

MOV bp, sp

; naudojamu registru issaugojimas

PUSHA

; rezultato eilute uzpildome tarpais

MOV cx, 6

MOV bx, [bp+4]

tarp: MOV byte ptr[bx], ' '

INC bx

LOOP tarp

; skaicius paruosiamas dalybai is 10

MOV ax, [bp+6]

MOV si, 10

val: XOR dx, dx

DIV si

; gauta liekana verciame i ASCII koda

ADD dx, '0' ; galima--> ADD dx, 30h

; irasome skaitmeni i eilutes pabaiga

DEC bx

MOV [bx], dl

; skaiciuojame pervestu simboliu kieki

INC cx

; ar dar reikia kartoti dalyba?

CMP ax, 0

JNZ val

POPA

POP bp

RET

binasc ENDP

prog ENDS

END pr