**1** cумма цифр

.model small

.stack

.data

x dw 989

.code

start:

mov ax,@data

mov ds,ax

mov bp,10

mov bh,0

mov ax,x

mov cx,3

a1: mov dx,0

div bp

add bh,dl

loop a1

mov al,bh

aam

add ax,3030h

mov dl,ah

mov dh,al

mov ah,2

cmp dl,'0'

jne m2

jmp m1

m2: int 21h

m1: mov dl,dh

int 21h

mov ah,0

int 16h

mov ah,4Ch

int 21h

end start

**2** вывод табл кодов

.model small

.stack

.data

.code

start:

mov ax,@data

mov ds,ax

mov cx,255

a1: mov bx,255

sub bx,cx

mov dx,bx

mov ah,2

int 21h

mov dl,'-'

int 21h

push cx

call print

pop cx

loop a1

mov dl,255

mov ah,2

int 21h

mov dl,'-'

int 21h

mov bx,255

call print

mov ah,0

int 16h

mov ah,4Ch

int 21h

print proc

mov ax,bx

xor cx,cx

mov bx,10

cicl1:

xor dx,dx

div bx

push dx

inc cx

test ax,ax

jnz cicl1

mov ah,2

cicl2:

pop dx

add dl,'0'

int 21h

loop cicl2

mov dl,' '

int 21h

ret

end proc

end start

**3** вывод индексов чисел

.model small

.stack

.data

x dw 5879

y db 'None',13,10,'$'

z db '',13,10,'$'

.code

start:

mov ax,@data

mov ds,ax

mov bp,10

mov bh,0

mov ah,01h

int 21h

sub al,30h

mov bl,al

lea dx,z

mov ah,9

int 21h

mov ax,x

mov cx,4

a1: mov dx,0

div bp

cmp bl,dl

jne l1

mov bh,cl

l1: loop a1

cmp bh,0

je met1

mov dl,bh

add dl,'0'

mov ah,2

int 21h

jmp end1

met1:

lea dx,y

mov ah,9

int 21h

end1:

mov ah,0

int 16h

mov ah,4Ch

int 21h

end start

**4-5** задом наперёд

.model small

.stack

.data

x dw 1234

.code

start:

mov ax,@data

mov ds,ax

mov bx,10

mov ax,x

mov cx,4

a1: xor dx,dx

div bx

add dl,'0'

push ax

mov ah,2

int 21h

pop ax

loop a1

mov ah,0

int 16h

mov ah,4Ch

int 21h

end start

**6** сумма двух первых и сумма двух вторых

.model small

.stack

.data

x dw 9897

str1 db 'Not equal $'

str2 db 'Equal $'

.code

start:

mov ax,@data

mov ds,ax

mov bp,10

mov bx,0

mov ax,x

mov cx,4

loop1:

mov dx,0

div bp

cmp cx,2

jbe met1

add bh,dl

loop loop1

jmp print

met1:

add bl,dl

loop loop1

print:

mov ah,9

cmp bh,bl

jne m1

mov dl,offset str2

int 21h

jmp m2

m1: mov dl,offset str1

int 21h

m2: mov ah,0

int 16h

mov ah,4Ch

int 21h

end start

**7** вывод натуральных чисел из строки

model small

.stack

.data

mess db "Vvedite stroku: ",0Dh,0Ah,'$'

pryzok db "",0Dh,0Ah,'$'

str1ml db 200

str1l DB '$'

str1 DB 200 dup('$')

str2 DB 200 dup('$')

.code

ASSUME DS:@data, es:@data

main:

mov ax,@data

mov ds,ax

mov es,ax

xor ax,ax

xor bx,bx

mov bh,2

lea dx,mess

mov ah,09h

int 21h

lea dx,str1ml

mov ah,0ah

int 21h

lea dx,pryzok

mov ah,09h

int 21h

xor cx,cx

mov cl,str1l

cld

lea si,str1

lea di,str2

xor ax,ax

cycle1:

lods str1

cmp al,'0'

jb m1

cmp al,'9'

ja m1

stos str2

loop cycle1

jmp vyvod

m1:

loop cycle1

vyvod:

lea dx,str2

mov ah,9

int 21h

mov ah,0

int 16h

mov ah,4ch

int 21h

end main

**8** сумма двух первых больше или меньше

.model small

.stack

.data

x dw 5432

str1 db 'Greater$'

str2 db 'Less$'

str3 db 'Equal$'

.code

start:

mov ax,@data

mov ds,ax

mov bp,10

mov bx,0

mov ax,x

mov cx,4

loop1:

mov dx,0

div bp

cmp cx,2

jbe met1

add bh,dl

loop loop1

jmp print

met1:

add bl,dl

loop loop1

print:

mov ah,9

cmp bl,bh

jae m1

mov dl,offset str2

int 21h

jmp m3

m1: cmp bl,bh

ja m2

mov dl,offset str3

int 21h

jmp m3

m2: mov dl,offset str1

int 21h

m3: mov ah,0

int 16h

mov ah,4Ch

int 21h

end start

**9** найти максимальную цифру

.model small

.stack

.data

x dw 5879

.code

start:

mov ax,@data

mov ds,ax

mov bp,10

mov bh,0

mov ax,x

mov cx,4

a1: mov dx,0

div bp

cmp bh,dl

jge l1

mov bh,dl

l1: loop a1

mov al,bh

aam

add ax,3030h

mov dl,ah

mov dh,al

mov ah,2

cmp dl,'0'

jne m2

jmp m1

m2: int 21h

m1: mov dl,dh

int 21h

mov ah,0

int 16h

mov ah,4Ch

int 21h

end start

**10** вывод 2 символов с макс и мин ascii

.model small

.stack

.data

mess db "Vvedite stroku: ",0Dh,0Ah,'$'

pryzok db "",0Dh,0Ah,'$'

str1ml db 200

str1l DB '$'

str1 DB 200 dup('$')

.code

main:

mov ax,@data

mov ds,ax

mov es,ax

xor ax,ax

xor bx,bx

mov bl,255

lea dx,mess

mov ah,09h

int 21h

lea dx,str1ml

mov ah,0ah

int 21h

lea dx,pryzok

mov ah,09h

int 21h

xor cx,cx

mov cl,str1l

cld

lea si,str1

xor ax,ax

cycle1:

lods str1

cmp bh,al

jae met1

mov bh,al

met1:

cmp bl,al

jbe met2

mov bl,al

loop cycle1

jmp vyvod

met2:

loop cycle1

vyvod:

mov dl,bl

mov ah,2

int 21h

mov dl,bh

int 21h

exit:

mov ah,0

int 16h

mov ah,4ch

int 21h

end main

**11** слож 2 неупакован

.model small

.stack

.data

num1 db 2,4,6,9,0

len1 = $ - num1

num2 db 7,6,4,0,0

rezult db len1 dup(0)

.code

start:

mov ax,@data

mov ds,ax

xor si,si

mov cx,len1

summ:

mov al,num1[si]

adc al,num2[si]

aaa

mov rezult[si],al

inc si

loop summ

mov cx,si

dec si

print:

mov dl,rezult[si]

add dl,'0'

mov ah,2

int 21h

dec si

loop print

mov ah,0

int 16h

mov ah,4Ch

int 21h

end start

**12** вычетание неупакован

.model small

.stack

.data

num1 db 2,4,6,9,0

len1 = $ - num1

num2 db 7,6,4,0,0

rezult db len1 dup(0)

.code

start:

mov ax,@data

mov ds,ax

xor si,si

mov cx,len1

summ:

mov al,num1[si]

sbb al,num2[si]

aas

mov rezult[si],al

inc si

loop summ

mov cx,si

dec si

print:

mov dl,rezult[si]

add dl,'0'

mov ah,2

int 21h

dec si

loop print

mov ah,0

int 16h

mov ah,4Ch

int 21h

end start

**13** умножение неупакован

.model small

.stack

.data

num1 db 2,5,3,0

len1 = $ - num1

num2 db 7

rezult db len1 dup(0)

.code

start:

mov ax,@data

mov ds,ax

xor si,si

xor dx,dx

mov cx,len1

summ:

mov al,num1[si]

mul num2

aam

adc al,dl

aaa

mov dl,ah

mov rezult[si],al

inc si

loop summ

mov cx,si

dec si

print:

mov dl,rezult[si]

add dl,'0'

mov ah,2

int 21h

dec si

loop print

mov ah,0

int 16h

mov ah,4Ch

int 21h

end start

**14** дел неупакован

.model small

.stack

.data

x db 4,7,3

y db 2

len equ 3

deln db 6 dup(0)

.code

start:

mov ax,@data

mov ds,ax

xor ax,ax

xor dx,dx

xor bx,bx

xor si,si

mov cx,len

delit:

xor ax,ax

mov al,x[si]

mov ah,dl

aad

div y

mov deln[si],al

mov dl,ah

inc si

loop delit

mov ah,2

mov bx,si

tost:

dec si

mov al,deln[si]

push ax

cmp si,0

jnz tost

vyvod:

dec bx

pop dx

add dl,30h

int 21h

cmp bx,0

jnz vyvod

mov ah,0

int 16h

mov ah,4ch

int 21h

end start

**15** сложение упакован

.model small

.stack

.data

x db 8h

y db 1h

sum db 3 dup(0)

.code

start:

mov ax,@data

mov ds,ax

xor ax,ax

mov bl,16

xor dx,dx

xor si,si

mov al,x

add al,y

daa

div bl

mov sum[si],ah

inc si

mov sum[si],al

inc si

jnc m2

mov al,0

adc al,0

mov sum[si],al

inc si

m2:

dec si

mov ah,2

mov dl,sum[si]

add dl,30h

int 21h

cmp si,0

jne m2

mov ah,0

int 16h

mov ah,4ch

int 21h

end start

**16** вычетание упакован

.model small

.stack

.data

x db 99h

y db 4h

sum db 3 dup(0)

.code

start:

mov ax,@data

mov ds,ax

xor ax,ax

mov bl,16

xor dx,dx

xor si,si

mov al,x

sub al,y

das

div bl

mov sum[si],ah

inc si

mov sum[si],al

inc si

jnc m2

sbb al,0

mov sum[si],al

inc si

m2:

dec si

mov ah,2

mov dl,sum[si]

add dl,30h

int 21h

cmp si,0

jne m2

mov ah,0

int 16h

mov ah,4ch

int 21h

end start

**17** выв\_кол\_вхожд\_символа\_в\_строке

.model small

.stack

.data

mess1 db "Vvedite simvol: ",0Dh,0Ah,'$'

mess2 db "Vvedite stroku: ",0Dh,0Ah,'$'

pryzok db "",0Dh,0Ah,'$'

str1ml db 200

str1l DB '$'

str1 DB 200 dup('$')

.code

main:

mov ax,@data

mov ds,ax

mov es,ax

xor ax,ax

xor bx,bx

lea dx,mess1

mov ah,09h

int 21h

mov ah,01h

int 21h

mov bl,al

lea dx,pryzok

mov ah,09h

int 21h

lea dx,mess2

mov ah,09h

int 21h

lea dx,str1ml

mov ah,0ah

int 21h

lea dx,pryzok

mov ah,09h

int 21h

xor cx,cx

mov cl,str1l

cld

lea si,str1

xor ax,ax

cycle1:

lods str1

cmp bl,al

je met1

loop cycle1

jmp vyvod

met1:

inc bh

loop cycle1

vyvod:

mov dl,bh

add dl,'0'

mov ah,2

int 21h

mov ah,0

int 16h

mov ah,4ch

int 21h

end main

**18** Строка символов в обр\_порядке

.model small

.stack

.data

mess db "Vvedite stroku: ",0Dh,0Ah,'$'

pryzok db "",0Dh,0Ah,'$'

str1ml db 200

str1l DB '$'

str1 DB 200 dup('$')

.code

main:

mov ax,@data

mov ds,ax

mov es,ax

xor ax,ax

lea dx,mess

mov ah,09h

int 21h

lea dx,str1ml

mov ah,0ah

int 21h

lea dx,pryzok

mov ah,09h

int 21h

xor cx,cx

mov cl,str1l

cld

lea si,str1

xor ax,ax

cycle1:

lods str1

push ax

loop cycle1

zapis:

mov cl,str1l

xor dx,dx

cycle2:

pop ax

mov ah,02h

mov dx,ax

int 21h

loop cycle2

exit:

mov ah,0

int 16h

mov ah,4ch

int 21h

end main

**19**выв\_четные\_элементы

.model small

.stack

.data

mess db "Vvedite stroku: ",0Dh,0Ah,'$'

pryzok db "",0Dh,0Ah,'$'

str1ml db 200

str1l DB '$'

str1 DB 200 dup('$')

str2 DB 200 dup('$')

.code

ASSUME DS:@data, es:@data

main:

mov ax,@data

mov ds,ax

mov es,ax

xor ax,ax

xor bx,bx

mov bh,2

lea dx,mess

mov ah,09h

int 21h

lea dx,str1ml

mov ah,0ah

int 21h

lea dx,pryzok

mov ah,09h

int 21h

xor cx,cx

mov cl,str1l

mov dl,cl

cld

lea si,str1

lea di,str2

xor ax,ax

cycle1:

mov al,dl

sub al,cl

div bh

cmp ah,0

jne m1

lods str1

loop cycle1

jmp vyvod

m1: lods str1

stos str2

loop cycle1

jmp vyvod

vyvod:

lea dx,str2

mov ah,9

int 21h

mov ah,0

int 16h

mov ah,4ch

int 21h

end main

**20**выв\_нечетные элементы

.model small

.stack

.data

mess db "Vvedite stroku: ",0Dh,0Ah,'$'

pryzok db "",0Dh,0Ah,'$'

str1ml db 200

str1l DB '$'

str1 DB 200 dup('$')

str2 DB 200 dup('$')

.code

ASSUME DS:@data, es:@data

main:

mov ax,@data

mov ds,ax

mov es,ax

xor ax,ax

xor bx,bx

mov bh,2

lea dx,mess

mov ah,09h

int 21h

lea dx,str1ml

mov ah,0ah

int 21h

lea dx,pryzok

mov ah,09h

int 21h

xor cx,cx

mov cl,str1l

mov dl,cl

cld

lea si,str1

lea di,str2

xor ax,ax

cycle1:

mov al,dl

sub al,cl

div bh

cmp ah,0

je m1

lods str1

loop cycle1

jmp vyvod

m1: lods str1

stos str2

loop cycle1

jmp vyvod

vyvod:

lea dx,str2

mov ah,9

int 21h

mov ah,0

int 16h

mov ah,4ch

int 21h

end main

**21**Упорядочен виде порядке

.model small

.stack 100h

.data

msg1 db 13,10,'Vvedite $'

msg2 db 13,10,'Result $'

buf db 20,21 dup('$')

.code

main:

mov ax,@data

mov ds,ax

mov ah,9h

lea dx,msg1

int 21h

lea dx,buf

mov ah,0Ah

int 21h

xor cx,cx

mov cl,buf[1]

xor si,si

xor di,di

mov si,2

dec cx

cycle1:

mov di,si

inc di

mov al,buf[si]

mov bl,cl

cycle2:

cmp al,buf[di]

jl met

mov dl,al

mov al,buf[di]

mov buf[di],dl

mov buf[si],al

met:

inc di

loop cycle2

mm:

mov cl,bl

inc si

loop cycle1

mov ah,9h

lea dx,msg2

int 21h

lea dx,buf[2]

int 21h

mov ah,4ch

int 21h

end main

**22**выв\_стрроку\_исключ\_буквы

.model small

.stack

.data

mess db "Vvedite stroku: ",0Dh,0Ah,'$'

pryzok db "",0Dh,0Ah,'$'

str1ml db 200

str1l DB '$'

str1 DB 200 dup('$')

str2 DB 200 dup('$')

.code

ASSUME DS:@data, es:@data

main:

mov ax,@data

mov ds,ax

mov es,ax

xor ax,ax

xor bx,bx

mov bh,2

lea dx,mess

mov ah,09h

int 21h

lea dx,str1ml

mov ah,0ah

int 21h

lea dx,pryzok

mov ah,09h

int 21h

xor cx,cx

mov cl,str1l

cld

lea si,str1

lea di,str2

xor ax,ax

cycle1:

lods str1

cmp al,'A'

jb m1

cmp al,'Z'

ja m1

loop cycle1

jmp vyvod

m1:

cmp al,'a'

jb m2

cmp al,'z'

ja m2

loop cycle1

jmp vyvod

m2:

stos str2

loop cycle1

vyvod:

lea dx,str2

mov ah,9

int 21h

mov ah,0

int 16h

mov ah,4ch

int 21h

end main

**23**совпадающие символы

.model small

.stack

.data

mes\_vvod db 13,10,'Vvedite stroku: $'

mes\_rez db 13,10,'Rezult: $'

buf1 db 20

len1 db 0

str1 db 20 dup('$')

buf2 db 20

len2 db 0

str2 db 20 dup('$')

.code

;ASSUME ds:@data,es:@data

main:

mov ax,@data

mov ds,ax

mov es,ax

xor ax,ax

CLD

lea dx,mes\_vvod

mov ah,09h

int 21h

lea dx,buf1

mov ah,0ah

int 21h

lea dx,mes\_vvod

mov ah,09h

int 21h

lea dx,buf2

mov ah,0ah

int 21h

lea di,str1

lea si,str2

xor ax,ax

xor dx,dx

xor cx,cx

mov cl,len1

lea dx,mes\_rez

mov ah,09h

int 21h

cycle1:

jcxz exit

repne cmpsb

jne cycle1

dec si

lodsb

mov dx,ax

mov ah,02h

int 21h

jmp cycle1

exit:

mov ax,4c00h

int 21h

end main

**24**Удалить символ второй строки из первой

.model small

.stack 100h

.data

msg1 db 13,10,'Vvedite stroku $'

msg2 db 13,10,'Result $'

msg3 db 'Vse sovpalo. Udalena stroka$'

str1 db 20,21 dup('$')

str2 db 20,21 dup('$')

res db 20 dup ('$')

.code

main:

mov ax,@data

mov ds,ax

mov es,ax

cld

mov ah,9h

lea dx,msg1

int 21h

lea dx,str1

mov ah,0Ah

int 21h

mov ah,9h

lea dx,msg1

int 21h

mov ah,0Ah

lea dx,str2

int 21h

xor cx,cx

mov cl,str1[1]

xor si,si

xor di,di

lea si,str1[2]

lea di,str2[2]

xor bx,bx

xor ax,ax

cycle:

lodsb

mov dl,cl

mov cl,str2[1]

repne scasb

jcxz met

jmp mm:

met:

mov res[bx],al

inc bx

mm:

mov cl,dl

loop cycle

mov ah,9h

cmp bl,0

jne print\_

lea dx,msg3

int 21h

jmp exit\_

print\_:

lea dx,msg2

int 21h

lea dx,res

int 21h

exit\_:

mov ah,4ch

int 21h

**25**Склеить две строки

.model small

.stack

.data

new db 13,10,'Novaya stroka: $'

req1 db 'Vvedite 1 stroku: $',13,10

req2 db 13,10,'Vvedite 2 stroku: $',13,10

buf1 db 5,5 dup(0)

buf2 db 5,5 dup(0)

new\_str db 14 dup(0)

len db 14

.code

assume ds:@data,es:@data

start:

mov ax,@data

mov ds,ax

mov es,ax

mov ah,9

lea dx,req1

int 21h

mov ah,0ah

lea dx,buf1-1

int 21h

mov cl,buf1

mov bl,cl

mov ah,9

lea dx,req2

int 21h

mov ah,0ah

lea dx,buf2-1

int 21h

mov cl,buf2

mov bh,cl

cld

lea si,buf1+1

lea di,new\_str

mov cl,bl

rep movs new\_str,buf1

mov cl,bh

lea si,buf2+1

rep movs new\_str,buf2

mov ah,9

lea dx,new

int 21h

mov cl,len

lea si,new\_str

m1:

lods new\_str

mov ah,2

mov dl,al

int 21h

loop m1

mov ah,0

int 16h

mov ax,4ch

int 21h

end start

**26**вернуть ответ через стек

.model tiny

.stack

.data

a dw 5

b dw 3

result dw ?

.code

mulproc proc near

push bp

mov bp, sp

mov ax, [bp+4]

mov dx, [bp+8]

mov bx, [bp+12]

mul dx

mov [bx], ax

mov sp, bp

pop bp

ret

mulproc endp

start:

mov ax, @data

mov ds, ax

push offset result

push dword ptr[a]

push dword ptr[b]

call mulproc

mov ax, [result]

aam

add ax, 3030h

mov dl, ah

mov dh, al

mov ah, 2

int 21h

mov dl, dh

int 21h

mov ah, 4ch

int 21h

end start

**27**вернуть ответ через регистр

.model tiny

.stack

.code

sumproc proc near

push bp

mov bp, sp

mov ax, [bp+4]

mov dx, [bp+6]

add ax, dx

mov sp, bp

pop bp

ret

sumproc endp

start:

push 5

push 3

call sumproc

aam

add ax, 3030h

mov dl, ah

mov dh, al

mov ah, 2

int 21h

mov dl, dh

int 21h

mov ah, 4ch

int 21h

end start

**28**вернуть ответ через общую память

.model small

.stack

.data

CR equ 0Dh

LF equ 0Ah

EOS equ '$'

crlf db CR, LF, EOS

a db 2,0,2 dup(0)

b db 2,0,2 dup(0)

c db 1 dup(0)

res db 2 dup(0), EOS

prmpt db 'vvedite 4islo 1: ', EOS

prmpt2 db 'vvedite 4islo 2: ', EOS

prmpt3 db 'summa: ', EOS

.code

myproc proc near

push bp

mov bp,sp

mov al, [bp+4]

add al, [bp+6]

sub al, 96

mov [c], al

mov sp,bp

pop bp

ret

endp myproc

start:

mov ax, @data

mov ds, ax

lea dx, prmpt

mov ah, 9

int 21h

mov ah, 0Ah

lea dx, a

int 21h

lea dx, crlf

mov ah, 9

int 21h

lea dx, prmpt2

mov ah, 9

int 21h

mov ah, 0Ah

lea dx, b

int 21h

lea dx, crlf

mov ah, 9

int 21h

xor ax, ax

mov al, [a+2]

push ax

mov al, [b+2]

push ax

call myproc

mov al, [c]

aam

add ah, 48

mov [res], ah

add al, 48

mov [res+1], al

lea dx, prmpt3

mov ah, 9

int 21h

lea dx, res

mov ah, 9

int 21h

mov ah, 4ch

int 21h

end start

**29** MessageBox

.386

.model flat

includelib import32.lib

extrn ExitProcess:PROC

extrn MessageBoxA:PROC

.data

Text db "First ASSEMBLER program",0h

Caption db 'Hello, World!!!!',0h

.code

start:

push 14h ;nabor knopok

push offset Caption ;zagolovok okna

push offset Text ;Text v okne

push 0h ;Vladelets okna

call MessageBoxA

push 0h

call ExitProcess

end start

**30** кос\_угла

#include "stdafx.h"

#include <iostream>

int \_tmain(int argc, \_TCHAR\* argv[])

{

float a = 1;

float result = 0;

\_\_asm{

finit

fld a

fcos

fst result

}

std::cout<<result;

std::cin.get();

return 0;

}

**31** танг\_угла

// 32.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

int \_tmain(int argc, \_TCHAR\* argv[])

{

float x=34;

float result=0;

\_asm{

finit

fld x

fptan

fstp result

fstp result

}

std::cout<<result;

std::cin.get();

return 0;

}

**32**длина\_окружности

#include "stdafx.h"

#include <iostream>

int \_tmain(int argc, \_TCHAR\* argv[])

{

float radius = 0.5;

float two = 2;

float result = 0;

\_asm{

finit

fld radius

fld radius

fmul

fldpi

fmul

fst result

}

std::cout<<result;

std::cin.get();

return 0;

}

**33**определ\_длину\_гипотенузы

// 23.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

int \_tmain(int argc, \_TCHAR\* argv[])

{

float a=5;

float b=7;

float two=2;

float result=0;

\_asm{

finit

fld a

fmul a

fld b

fmul b

fadd

fsqrt

fst result

}

std::cout<<result;

std::cin.get();

return 0;

}

**34** (x-z)\*(2х-z)

// 24.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

int \_tmain(int argc, \_TCHAR\* argv[])

{

float two = 2;

float x = 6;

float z = 2;

float y = 0;

\_asm{

finit

fld two

fld z

fld x

fmul two

fsub

fld z

fsub x

fmul

fst y

}

std::cout<<y;

std::cin.get();

return 0;

}

**35**вывести\_простые\_числа\_до\_заданного

#include "stdafx.h"

#include <iostream>

void main()

{

float a = 0;

int max = 0;

int flag;

std::cin>>max;

for (float i = 2; i < max; i++) {

flag = 1;

for (float j = 2; j < i; j++)

{

\_asm{

finit

fld i

fdiv j

fst a

fld a

frndint

fsubr

fst a

}

if (a == 0){

flag = 0;

}

}

if (flag == 1){

std::cout<<i<<" prostoe.\n";

}

}

system("pause");

}

**36** (x\*e^x)/4

// 37.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include <iostream>

int \_tmain(int argc, \_TCHAR\* argv[])

{

float x=5;

float mn = 4;

float result = 0;

float cel = 0;

\_asm{

finit

fld x

fldl2e

fmul

fst result

fld result

frndint

fst cel

fsub

f2xm1

fld1

fadd

fst result

fld cel

fld result

fscale

fld mn

fdiv

fst result

}

std::cout<<result;

std::cin.get();

return 0;

}

**37** 135x+(cx-2042b)/a^2

// 38.cpp : Defines the entry point for the console application.

#include "stdafx.h"

#include <iostream>

int \_tmain(int argc, \_TCHAR\* argv[])

{

float x=4;

float mn1 = 135;

float mn2 = 2042;

float c = 3;

float a = 2;

float b = 5;

float result = 0;

float prom1 = 0;

float prom2 = 0;

float prom3 = 0;

\_asm{

finit

fld x

fld mn1

fmul

fst prom1

fld c

fld x

fmul

fst prom2

fld mn2

fld b

fmul

fld prom2

fsubr

fst prom2

fld a

fmul a

fdiv

fld prom1

fadd

fst result

}

std::cout<<result;

std::cin.get();

return 0;

}

**38**вывести\_знач\_пи\_с\_точностью\_5\_знаков

#include "conio.h"

#include <iostream>

void main()

{0000;

float y = 0;

float a = 10;

\_\_asm{

finit

fldpi

fmul a

fmul a

fmul a

fmul a

fmul a

frndint

fdiv a

fdiv a

fdiv a

fdiv a

fdiv a

fst y

}

std::cout<<y;

getch();

}