**Se foloseste o zona de memorie de 256 biti pentru a memora 32 de octeti...In R16 se da un numar intre 0..255. Scrieti o functie care returneaza in R18 valoarea bitului cu numarul (R16).**

.include "m32def.inc"

.dseg

S1: .byte 100

.cseg

ldi R18,0

ldi R16,low(RAMEND)

ldi R17,high(RAMEND)

out SPL,R16

out SPH,R17

ldi XL,low(S1)

ldi XH,high(S1)

ldi R20,0

ldi R21,0

SIR:

ldi R16,0b00000000

st X+,R16

ldi R16,0b00000000

st X+,R16

ldi R16,0b00000000

st X+,R16

ldi R16,0b11011111

st X,R16

ldi XL,low(S1)

ldi XH,high(S1)

ldi R16,26

**PROC1:**

**subi R16,8**

**inc R20**

**cpi R16,8**

**brsh PROC1**

**mov R21,R16**

**PROC2:**

**ld R18,X+**

**dec R20**

**brne PROC2**

**ldi R22,0b10000000**

**mov R25,R21**

**ldi R23,7**

**sub R23,R25**

**PROC3:**

**lsr R22**

**dec R21**

**brne PROC3**

**ld R18,X**

**and R18,R22**

**PROC4:**

**lsr R18**

**dec R23 ;=> in R18 se pune 0**

**brne PROC4**

**STOP:**

**rjmp STOP**

**Sa se numere cate cuvinte sunt intr-un sir.**

.include "m32def.inc"

.dseg

S1: .byte 20

.cseg

ldi R16,low(RAMEND)

ldi R17,high(RAMEND)

out SPL,R16

out SPH,R17

ldi XL,low(S1)

ldi XH,high(S1)

ldi R18,0

ldi R20,0

SIR: ldi R16,'C'

st X+,R16

ldi R16,'e'

st X+,R16

ldi R16,'l'

st X+,R16

ldi R16,' '

st X+,R16

ldi R16,'m'

st X+,R16

ldi R16,'a'

st X+,R16

ldi R16,'i'

st X+,R16

ldi R16,' '

st X+,R16

ldi R16,'m'

st X+,R16

ldi R16,'a'

st X+,R16

ldi R16,'r'

st X+,R16

ldi R16,'e'

st X+,R16

ldi R16,' '

st X+,R16

ldi R16,'l'

st X+,R16

ldi R16,'a'

st X+,R16

ldi R16,'c'

st X+,R16

ldi R16,' '

st X+,R16

ldi R16,'e'

st X+,R16

ldi R16,'s'

st X+,R16

ldi R16,'t'

st X+,R16

ldi R16,'e'

st X+,R16

ldi R16,' '

st X+,R16

ldi R16,'M'

st X+,R16

ldi R16,'a'

st X+,R16

ldi R16,'r'

st X+,R16

ldi R16,'e'

st X+,R16

ldi R16,'a'

st X+,R16

ldi R16,' '

st X+,R16

ldi R16,'C'

st X+,R16

ldi R16,'a'

st X+,R16

ldi R16,'s'

st X+,R16

ldi R16,'p'

st X+,R16

ldi R16,'i'

st X+,R16

ldi R16,'c'

st X+,R16

ldi R16,'a'

st X+,R16

ldi R16,'.'

st X,R16

ldi XL,low(S1)

ldi XH,high(S1)

rcall PROC1

**STOP:**

**rjmp STOP**

**PROC1:**

**ld R16,X+**

**cpi R16,' '**

**breq SPATIU**

**cpi R16,'.'**

**breq PUNCT**

**rjmp PROC1**

**SPATIU:**

**ldi R18,1**

**add R20,R18**

**clr R18**

**rjmp PROC1**

**PUNCT:**

**ldi R18,1**

**add R20,R18**

**ret**

**Avand in X adresa de inceput a unui sir de caractere, trebuia dat in Y acelasi sir dar caracterele TAB trebuiau inclocuite cu spatii.**

.include "m32def.inc"

.dseg

S1: .byte 100

S2: .byte 100

.cseg

ldi R16,low(RAMEND)

ldi R17,high(RAMEND)

out SPL,R16

out SPH,R17

ldi XL,low(S1)

ldi XH,high(S1)

SIR:

ldi R16,'C'

st X+,R16

ldi R16,'e'

st X+,R16

ldi R16,9

st X+,R16

ldi R16,'m'

st X+,R16

ldi R16,'a'

st X+,R16

ldi R16,'i'

st X+,R16

ldi R16,9

st X+,R16

ldi R16,'f'

st X+,R16

ldi R16,'a'

st X+,R16

ldi R16,'c'

st X+,R16

ldi R16,'i'

st X+,R16

ldi R16,'?'

st X,R16

ldi XL,low(S1)

ldi XH,high(S1)

ldi YL,low(S2)

ldi YH,high(S2)

rcall PROC1

**STOP:**

**rjmp STOP**

**PROC1:**

**ld R16,X+**

**cpi R16,9**

**breq INLOC**

**cpi R16,'?'**

**breq SFARSIT**

**st Y+,R16**

**rjmp PROC1**

**INLOC:**

**ldi R16,' '**

**st Y+,R16**

**rjmp PROC1**

**SFARSIT:**

**st Y,R16**

**ret**

**Scrieti o procedura care primeste in X adresa unui sir de caractere si intoarce in R20 nr. de caractere aflate intre paranteze.**

.include "m32def.inc"

.dseg

S1: .byte 20

.cseg

ldi R16,low(RAMEND)

ldi R17,high(RAMEND)

out SPL,R16

out SPH,R17

ldi XL,low(S1)

ldi XH,high(S1)

ldi R18,0

ldi R20,0

SIR:

ldi R16,'a'

st X+,R16

ldi R16,'b'

st X+,R16

ldi R16,'('

st X+,R16

ldi R16,'c'

st X+,R16

ldi R16,'d'

st X+,R16

ldi R16,')'

st X+,R16

ldi R16,'e'

st X+,R16

ldi R16,'f'

st X+,R16

ldi R16,'g'

st X+,R16

ldi R16,'('

st X+,R16

ldi R16,'h'

st X+,R16

ldi R16,'i'

st X+,R16

ldi R16,'j'

st X+,R16

ldi R16,')'

st X+,R16

ldi R16,'1'

st X+,R16

ldi R16,'.'

st X,R16

ldi XL,low(S1)

ldi XH,high(S1)

rcall PROC1

**PROC1:**

**ld R16,X+**

**cpi R16,'.'**

**breq GATA**

**cpi R16,'('**

**breq DESCHISA**

**cpi R16,')'**

**breq INCHISA**

**add R20,R18**

**rjmp PROC1**

**GATA: ret**

**DESCHISA:**

**ldi R18,1**

**rjmp PROC1**

**INCHISA:**

**ldi R18,0**

**rjmp PROC1**

**Scrieti o procedura ce calculeaza in bitul 0 din R17 urmatoarea expresie : [R16.0 si (R16.1 sau R16.2)]**

EXPRESIE:

mov R17,R16

lsr R17 ; R17.2 -> R17.1

or R17,R16 ; R17.1=R16.1 sau R16.2

mov R18, R16

lsl R18 ; R18.0 -> R18.1

and R18,R17 ; R18.1=expresia

ldi R17,0

andi R18,0b00000010

breq EXPRESIE1

ldi R17,0b00000001

EXPRESIE1: ret

SAU:

EXPRESIE:

clr R17

mov R18,R16

lsr R18

or R18,R16

lsr R18

mov R17,R16

and R17,R18

andi R17,0b00000001

**Sa se numere cate grupuri de cel putin 2 biti de 1 consecutivi exista in ...**

.include "m32def.inc"

.dseg

.cseg

clr R20

clr R21

clr R22

;initializare sir

ldi XL,0b00001111

ldi XH,0b00000000

;initializare masca

ldi YL,0b00000001

ldi YH,0b00000000

ldi R16,0b00000000

ldi R17,0b00000000

PROC1:

movw Z,X

and ZL,YL

and ZH,YH

cp ZL,R16

cpc ZH,R17

brne PROC2

rcall ROTIRE1

ROTIRE1:

clr R20

lsl YL

rol YH

inc R22

cpi R22,16

breq STOP

rcall PROC1

ROTIRE:

lsl YL

rol YH

inc R22

cpi R22,16

breq STOP

rcall PROC1

PROC2:

inc R20

cpi R20,2

breq PROC3

rcall ROTIRE

PROC3:

inc R21

rcall ROTIRE

STOP:

Ret

**Se da in X un numar intre 1 si 365. Sa se scrie in R20 in ce luna din an se afla ziua cu nr. (X) si in R21 numarul de ordine in cadrul lunii.**

.include "m32def.inc"

.cseg

ldi XL,low(340)

ldi XH,high(340)

DECEMBRIE:

ldi R17,low(335)

ldi R16,high(335)

cp XL,R17

cpc XH,R16

brlo IANUARIE

ldi R20,12

mov YL,XL

mov YH,XH

adiw X,1

subi XL,low(335)

sbci XH,high(335)

IANUARIE: ret

.

.

.

.