

Задачи по КПО

1) Дан массив целых знаковых чисел и целое знаковое число, посчитать кол-во элементов массива больше чем число.

Arr dw 5, 1, 2, 3, 4, 5

Chislo dw 3

Push Arr

Push [Chislo]

Call Procedure

Procedure:

Push bp

mov bp, sp

Push bx

mov bx, [bp + 6]

Xor ax, ax

mov cx, [bx]

mov dx, [bp + 4]

@@:

add bx, 2

cmp word[bx], dx

jle Skip

inc ax

Skip:

Loop @B

Pop bx bp

Ret 4

2) Дан массив, найти кол-во элементов, у которых старшие 2 бита единицы

Arr dw 4, 1, 2, 3, 4

Push Arr

Call Procedure

Procedure:

```
    Push bp
    mov bp, sp
    Push bx
    mov bx, [bp + 4]
    mov cx, word [bx]
    Xor ax, ax
    Push si
```

@@:

```
    add bx, 2
    Xor dx, dx
    mov si, word [bx]
    Shl si, 1
    Adc dx, 0
    Shl si, 1
    Adc dx, 0
    Cmp dx, 2
    jne Skip
    inc ax
```

Skip:

```
    Loop @B
```

Pop si

Pop bx

Pop bp

Ret 2

```
//Для знаковых
jne = jump not equal
je = jump equal
jle = jump less equal
jge = jump greater equal
//Для беззнаковых
jb = jump below
ja = jump above
```

3) Сумма элементов массива

```
Arr dw 5, 1, 2, 3, 4, 5
```

```
Push Arr
Call Procedure
```

Procedure:

```
    Push bp
    mov bp, sp
    Push bx
    mov bx, [bp + 4]
    mov cx, word[bx]
    Xor ax, ax
```

```
@@:
    add bx, 2
    add ax, word[bx]
    Loop @B
```

```
Pop bx
Pop bp
Ret 2
```

4) Минимальный элемент массива

Arr 5, 1, 2, 3, 4, 5

Push Arr

Call Procedure

Procedure:

Push bp

mov bp, sp

Push bx

mov bx, [bp + 4]

mov cx, Word[bx]

add bx, 2

mov ax, Word[bx]

dec cx

@@:

add bx, 2

cmp word[bx], ax

jge Skip

mov ax, word[bx]

Skip:

Loop @B

Pop bx

Pop bp

Ret 2

5) Инvertировать число типа 11010 -> 01011

Num dw 10

push [Num]

call Procedure

Procedure:

```
push bp
mov bp, sp
mov dx, [bp + 4]
xor ax, ax
```

@@:

```
shl ax, 1 ;?
shr dx, 1
adc ax, 0
test dx, dx
jnz @B
```

```
pop bp
ret 2
```

6)Даны 2 массива, поменять местами элементы каждого индекса

```
xchg Word[bx], Word[si]
mov Word[bx], Word[si]
cmp, test...
```

//Два операнда в памяти быть не могут с данными инструкциями.

Arr1 dw 5, 1, 2, 3, 4, 5

Arr2 dw 5, 3, 4, 5, 6, 7

Push Arr1

Push Arr2

Call Procedure

Procedure:

```
Push bp
mov bp, sp
Push bx
mov bx, [bp + 4]
Push si
```

```
mov si, [bp + 6]
mov cx, Word[si]
```

@@:

```
add bx, 2
add si, 2
```

```
mov ax, Word[bx]
xchg ax, Word[si]
mov Word[bx], ax
```

Loop @B

```
Pop si bx bp
Ret 4
```

7) Дан массив и число N, нужно заполнить массив числами N-1, N-2, N-3 и т.д.

Arr dw 5, 5 dup ?

Chislo dw 6

```
Push Arr
Push [Chislo]
```

Call Procedure

Procedure:

```
Push bp
mov bp, sp
Push bx
mov bx, [bp + 6]
Xor ax, ax
mov cx, [bx]
mov dx, [bp + 4]
```

@@:

```
add bx, 2
dec dx
mov word[bx], dx
Loop @B
```

Pop bx bp

Ret 4

8) Дан массив знаковых чисел и знаковое число X, найти количество чисел, которые нацело делятся на X

Arr dw 5, 1, 2, 3, 4, 5

Chislo dw 2

Push Arr

Push [Chislo]

Call Procedure

Procedure:

```
Push bp
mov bp, sp
Push bx
mov bx, [bp + 6]
Push si
Xor si, si
mov cx, word[bx]
```

@@:

```
add bx, 2
mov ax, word[bx]
div [bp + 4]
test dx, dx
jnz Skip
inc si
```

Skip:

Loop @B

```
mov ax, si  
Pop bx bp  
Ret 4
```

9) Дано число X, надо написать, сколько 1 в двоичном представлении этого числа

Chislo dw 6

Push [Chislo]

Call Procedure

Procedure:

```
Push bp  
mov bp, sp  
mov dx, [bp + 4]  
Xor ax, ax
```

@@:

CF = 1 или 0 бит после сдвига или переполнения

```
Shr dx, 1
```

CF = сдвинутым битом, 0 или 1

```
Adc ax, 0
```

ax = ax + 0 + CF(0/1)

```
test dx, dx
```

```
jnz @B
```

Pop bp

Ret 2

ax, dx, cx, sp - не работает с адресами

bx, di, si, bp - работает с адресами. Надо сохранять при запуске процедуры

10) Даны 2 C-строки, проверить, равны ли они. 1-True/ 0- False

```
Pascal db 5, 'Meow!'
C-String db 'Meow!', 0
```

```
Str1 db 'Meow', 0
Str2 db 'Gaf', 0
```

```
Push Str1
Push Str2
```

Call Procedure

Procedure:

```
    push bp
    mov bp, sp
    push di
    mov di, [bp + 6]
    push si
    mov si, [bp + 4]
    xor ax, ax
```

Start:

```
    cmp Byte[di], 0
    je  @Maybe
    cmp Byte[si], 0
    je  @End
    movzx dx, Byte[di]
    cmp  dx, Byte[si]
    jne @End
    inc  si
    inc  di
    jmp  Start
```

@Maybe:

```
    cmp Byte[si], 0
    jne @End
    inc ax
```

@End:

Pop si di bp

Ret 4

11) Развернуть массив на 26 элементов (число походу рандомное по условию будет)

Arr dw 26, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26

Push Arr

Call Procedure

Procedure:

Push bp

mov bp, sp

Push bx

mov bx, [bp + 4]

mov cx, 13

Push si

mov si, bx

add si, 54

@@:

add bx, 2

sub si, 2

mov ax, word[si]

xchg word[bx], ax

mov word[si], ax

Loop @B

Pop si bx bp

Ret 2

12)Дана С-строка, состоящая из букв латинского алфавита и пробелов, найти в ней кол-во слов (может быть несколько пробелов

в конце, в начале и между словами)

Str db 'Meow', 0

Push Str

Call Procedure

Procedure:

```
    Push bp
    mov bp, sp
    Xor ax, ax
    Push bx
    mov bx, [bp + 4]
```

@@:

```
    cmp Byte[bx], 0
    je    End
    cmp Byte[bx], ' '
    jne Skip
    inc bx
    jmp @B
```

Skip:

```
    inc ax
```

@Start:

```
    inc bx
    cmp Byte[bx], 0
    je    End
    cmp Byte[bx], ' '
    je    @B
    jmp   @Start
```

End:

```
    Pop bx bp
    Ret 2
```

13) Дана C-строка, определить, является ли строка записью десятичного числа (пробелы в начале и в конце допускаются)
Результат 1-True/0-False

```
Str db ' 123 ', 0
```

```
Push Str
```

```
Call Procedure
```

```
Procedure:
```

```
    Push bp
    mov bp, sp
    push si
    mov si, [bp + 4]
    xor ax, ax
```

```
@@:
```

```
    cmp Byte[si], 0
    je  @End
    cmp Byte[si], ' '
    jne @Start
    inc si
    jmp @B
```

```
@Start:
```

```
    cmp Byte[si], 0
    je  @Inc
    cmp Byte[si], '0'
    jb  @End
    cmp Byte[si], '9'
    ja  @End
    cmp Byte[si], ' '
    je  @F
    inc si
    jmp @Start
```

```
@@:
```

```
    cmp Byte[si], 0
```

```

        je @Inc
        cmp Byte[si], ' '
        jne @End
        inc si
        jmp @B
@Inc:
        inc ax

@End:
        pop si bp
        ret 2

```

14) Заменить первые N элементов на последовательность:
-1,0,1,-1,0,1 (у меня подозрение, что могут меняться
последовательности, снизу приведен код для 1,-1,0, но его изменить
можно под условие, просто переставив строки)

```

Arr dw 6, 1, 2, 3, 4, 5, 6
elem dw 4

```

```

Push Arr
Push [elem]

```

Call Procedure

Procedure:

```
    Push bp
    mov bp, sp
    Push bx
    mov bx, [bp + 6]
    mov cx, [bp + 4]
```

@@:

```
    add bx, 2
    mov Word[bx], -1
    dec cx
    test cx, cx
    je End
    add bx, 2
    mov Word[bx], 0
    dec cx
    test cx, cx
    je End
    add bx, 2
    mov Word[bx], 1
    dec cx
    test cx, cx
    je End
    jmp @B
```

End:

```
    Pop bx bp
    Ret 4
```

15) Дана строка S, содержащая латинские буквы, пробел и знаки препинания, а именно: , : ;
Найти число знаков препинания, после которых нет пробела

```
Str    db 'Meow :::', 0
```

Push Str
Call Procedure

Procedure:

```
push bp
mov bp, sp
push si
mov si, [bp + 4]
xor ax, ax
```

```
@@:
    cmp Byte[si], 0
    je  @End
    cmp Byte[si], ','
    jne @F
    jmp @StartCheck
```

```
@@:
    cmp Byte[si], ':'
    jne @F
    jmp @StartCheck
```

```
@@:
    cmp Byte[si], ';'
    jne @Skip
```

```
@StartCheck:
    cmp Byte[si + 1], ' '
    je  @Skip
    inc ax
```

```
@Skip:
    jmp @B
```

Pop si bp
Ret 2

16) Дан массив A[32] беззнаковых чисел и число X, посчитать кол-во элементов массива меньших чем X

Arr dw 32, 1, 2, 3, 4, 5 ...

Chislo dw 5

Push Arr

Push [Chislo]

Call Procedure

Procedure:

Push bp

mov bp, sp

Push bx

mov bx, [bp + 6]

Xor ax, ax

mov cx, Word[bx]

mov dx, [bp + 4]

@@:

add bx, 2

cmp word[bx], dx

jge Skip

inc ax

Skip:

Loop @B

Pop bx bp

Ret 4

17) Дана C-строка и символ, найти первое вхождение символа в строку (индексация с 1), если символ не найден, то вернуть 0

movzx dx, Byte[bx]

dh dl

movsx dx, Byte[bx]


```
dh dl
200 до 255 = 1xxxxxxb
dh = 11111111
dx = 11111111xxxxxxb
100 до 255 = 0xxxxxxb
dh = 00000000b
dx = 00000000xxxxxxb
```

```
Str db 'Meow', 0
Simbol db 'M'
```

```
Push Str
Push Simbol
```

Call Procedure

Procedure:

```
    Push bp
    mov bp, sp
    Xor ax, ax
    Push bx
    mov bx, [bp + 4]
    movzx dx, [bx]
    mov bx, [bp + 6]
```

@@:

```
    cmp Byte[bx], 0
    je    End
    cmp Byte[bx], dx
    je Calculate
    inc bx
    jmp @B
```

Calculate:

```
    sub bx, [bp + 6]
    inc bx
    mov ax, bx
```

End:

```
Pop bx bp
Ret 4
```

18) Дано целое десятичное число $X = 564$, найти десятичную сумму цифр этого числа

```
Chislo dw 564
```

```
Push [Chislo]
Call Procedure
```

Procedure:

```
Push bp
mov bp, sp
mov ax, [bp + 4]
Xor cx, cx
```

@@:

```
mov dx, 10
div dx
add cx, dx
test ax, ax
jnz @B
//В cx результат
```

```
Pop bp
ret 2
```

19) Дан массив на 25 беззнаковых элементов (индексация с 0) и беззнаковое число $N \leq 25$. Найти кол-во элементов в массиве, чей индекс строго меньше N и старший бит равен 1

```
Arr dw 5, 1, 2, 3, 4, 5
Chislo dw 3
```

```
Push Arr
Push [Chislo]
```

Call Procedure

Procedure:

```
Push bp
mov bp, sp
Push bx
mov bx, [bp + 6]
mov dx, [bp + 4]
xor ax, ax
```

@@:

```
add bx, 2
cmp ax, dx
jge @End
Test Word[bx], Word[bx]
Jns @Skip
inc ax
```

@Skip:

```
Loop @B
```

@End:

```
Pop bx bp
Ret 4
```

20) Дано целое знаковое число N. Сформировать массив A[48].
Следующим образом: A[0] = N, A[1] = N-1, A[2] = N-2 и т.д.

```
Arr dw 48, 48 dup ?
Chislo dw 56
```

```
Push Arr
Push [Chislo]
Call Procedure
```

Procedure:

```
Push bp
mov bp, sp
Push bx
mov bx, [bp + 6] ; Адрес на массив
mov dx, [bp + 4] ; Само число N
mov cx, Word[bx] ; Счетчик из Адреса bx
```

@@:

```
add bx, 2 ; Переход к 1 элементу
mov word[bx], dx
dec dx
Loop @B
```

Pop bx bp

Ret 4

21)Дана C-строка из пробелов и букв. Все первые строчные буквы после пробелов заменить на заглавные

Str db 'me ow!', 0

Push Str

Call Procedure

Procedure:

```
Push bp
mov bp, sp
Push bx
mov bx, [bp + 4]
```

@Start:

```
cmp Byte[bx], 0
je @End
cmp Byte[bx], ' '
jne @F
inc bx
jmp @Start
```

@@:

```
cmp Byte[bx], 'a'
```

```
jb    @NextIteration
cmp   Byte[bx], 'z'
ja     @NextIteration
sub   Byte[bx], 'a' - 'A'
```

@NextIteration:

```
inc bx
cmp   Byte[bx], 0
je     @End
cmp   Byte[bx], ' '
je     @Start
jmp    @NextIteration
```

@End:

```
Pop bx bp
Ret 2
```

22) Дан массив целых знаковых чисел и целое знаковое число,
посчитать кол-во элементов массива меньше чем число.

```
Arr dw 5, 1, 2, 3, 4, 5
Numb dw 3
```

```
Push Arr
Push [Numb]
Call Procedure
```

Procedure:

```
Push bp
mov bp, sp
Push bx
mov bx, [bp + 6]
mov dx, [bp + 4]
mov cx, Word[bx]
xor ax, ax
```

@@:

```
add bx, 2
```

```

        cmp Word[bx], dx
        jge @Skip
        inc ax
@Skip:
        Loop @B

@End:
        pop bx bp
        ret 4

```

23) Дан массив беззнаковых чисел A[44], посчитать сумму четных элементов

```

Arr dw 44, 1, 2, 3, 4, 5...
Push Arr
Call Procedure
Procedure:
        push bp
        mov bp, sp
        push bx
        mov bx, [bp + 4]
        xor ax, ax
        mov cx, Word[bx]

```

```

@@:
        add bx, 2
        test Word[bx], 1
        jnz @Skip
        inc ax
@Skip:
        Loop @B

```

```

pop bx bp
ret 2

```

24) Дана C-строка. Найти её длину

```
str db 'meow', 0
```

```
push str  
call Procedure
```

Procedure:

```
    push bp  
    mov bp, sp  
    push bx  
    mov bx, [bp + 4]  
    xor cx, cx
```

@@:

```
    cmp Byte[bx], 0  
    je @End  
    inc cx  
    jmp  @B
```

@End:

```
    pop bx bp  
    ret  2
```

25)Заменить все строчные буквы на заглавные. Pascal строка

```
Str db 6, 'string'
```

```
Push Str  
Call Procedure
```

Procedure:

```
    Push bp  
    mov bp, sp  
    Push bx  
    mov bx, [bp + 4]  
    movzx cx, byte[bx]
```

@@:

```
    inc bx  
    cmp  Byte[bx], 'a'
```

```
jb    @Skip
cmp   Byte[bx], 'z'
ja    @Skip
sub   Byte[bx], 'a' - 'A'
```

```
@Skip:
Loop @F
```

```
Pop bx bp
Ret 2
```

26)Найти количество элементов в массиве меньше/больше данного.

```
Arr dw 5, 1, 2, 3, 4, 5
Chislo dw 3
```

```
Push Arr
Push [Chislo]
```

```
Call Procedure
```

```
Procedure:
```

```
Push bp
mov bp, sp
Push bx
mov bx, [bp + 6]
Xor ax, ax
mov cx, Word[bx]
mov dx, [bp + 4]
```

```
@@:
add bx, 2
```



```
    cmp word[bx], dx
    je Skip
    inc ax
```

```
Skip:
    Loop @B
```

```
Pop bx bp
Ret 4
```

27)Инвертировать 3-ий бит во всех элементах массива.

```
Arr dw 5, 1, 2, 3, 4, 5
```

```
Push Arr
```

```
Call Procedure
```

```
Procedure:
```

```
    Push bp
    mov bp, sp
    Push bx
    mov bx, [bp + 4]
    mov cx, Word[bx]
```

```
@@:
    add bx, 2
    xor Word[bx], 8 ; 8 = 1000b
    Loop @B
```

```
Pop bx bp
Ret 2
```

28)Найти сумму элементов в массиве, меньших данного и кратных 4

Arr dw 5, 1, 2, 3, 4, 5
Chislo dw 4

Push Arr
Push [Chislo]
Call Procedure

Procedure:

Push bp
mov bp, sp
Push bx
mov bx, [bp + 6]
mov dx, [bp + 4]
mov cx, word[bx]
xor ax, ax

@@:

add bx, 2
cmp word[bx], dx
jnl @Skip
test Word[bx], 3 ;11b
jnz @Skip
add ax, word[bx]

@Skip:

Loop @B

pop bx bp
ret 4

29) Дан массив и число. Все элементы массива, меньшие данного, увеличить на их индекс

Arr dw 5, 1, 2, 3, 4, 5
Chislo dw 3

Push Arr

Push [Chislo]

Call Procedure

Procedure:

```
Push bp
mov bp, sp
Push bx
mov bx, [bp + 6]
mov cx, [bx]
mov dx, [bp + 4]
xor ax, ax
```

@@:

```
add bx, 2
cmp word[bx], dx
jge @Skip
add word[bx], ax
```

@Skip:

```
inc ax
Loop @B
```

Pop bx bp

Ret 4

30)Шифрование Цезаря. Дана строка, состоящая из букв латинского алфавита и пробелов. Буквы заменить, прибавляя к коду символа 3. Т.е. 'a' на 'd'

Str db 'meow', 0

Push Str

Call Procedure

Procedure:

```
Push bp
```

```
mov bp, sp
Push bx
mov bx, [bp + 4]
```

```
@@:
    cmp byte[bx], 'a'
    jb @Skip
    cmp byte[bx], 'z'
    ja @Skip
    add byte[bx], 3
    cmp Byte[bx], 'z'
    jna @Skip
    sub Byte[bx], 'z'
    add Byte[bx], 'a'
```

```
@Skip:
    inc bx
    cmp byte[bx], 0
    je @End
```

```
@End:
    Pop bx, bp
    Ret 2
```

31) Дан массив и число N. Заменить первые N элементов на 1,-1,1,-1 и т.д.

```
Arr dw 6, 1, 2, 3, 4, 5, 6
elem dw 5
```

```
Push Arr
Push [elem]
```

```
Call Procedure
```

```
Procedure:
    Push bp
```

```
mov bp, sp
Push bx
mov bx, [bp + 6]
mov cx, [bp + 4]
```

@@:

```
add bx, 2
mov Word[bx], 1
dec cx
test cx, cx
jz End
add bx, 2
mov Word[bx], -1
Loop @B (В 14 задаче было без Loop)
```

End:

```
Pop bx bp
Ret 4
```

32) Найти количество элементов массива A, в старшем бите которых 1

```
Arr dw 4, 1, 2, 3, 4
```

```
Push Arr
```

```
Call Procedure
```

```
Procedure:
```

```
Push bp
mov bp, sp
Push bx
mov bx, [bp + 4]
mov cx, word[bx]
Xor ax, ax
```

@@:

```
add bx, 2
mov dx, word[bx]
shl dx, 1
adc ax, 0
```

```
@Skip:
    Loop @B
```

```
Pop bx bp
Ret 2
```

33) Даны массивы A[50] и B[50], элементы которых целые знаковые числа. Записать в A[i] исходное значение A[i] + B[i], а в B[i] записать исходные A[i] - B[i]

```
Arr1 dw 50, 1, 2, 3, 4...
Arr2 dw 50, 1, 2, 3, 4...
```

```
Push Arr1
Push Arr2
```

Call Procedure

Procedure:

```
    Push bp
    mov bp, sp
    Push bx
    mov bx, [bp + 4] ; B Arr
    Push si
    mov si, [bp + 6] ; A Arr
```

```
@@:
    add bx, 2
    add si, 2
    mov ax, Word[si] ; A[i]
    add ax, Word[bx] ; A[i] + B[i]
    mov Word[si], ax
    sub ax, Word[bx] ; A[i]
```

```
sub ax, Word[bx] ; A[i] - B[i]  
mov Word[bx], ax  
Loop @B
```

```
Pop si bx bp  
Ret 4
```