

Задание 4

4.1.22. Написать полную программу решения задачи. Дана непустая последовательность символов из набора ['0'...'7'] – запись числа в 8-ой системе, за последовательностью следует пробел. Ввести это число и напечатать его в десятичной системе. Считать, что запись числа корректная и что само число умещается в двойное слово. Операцию mul не использовать.

```
Include console.inc
```

```
.code
```

```
Start:
```

```
Xor eax, eax
```

```
Xor ebx, ebx
```

```
L:
```

```
Inchar bl, ''
```

```
Je L1
```

```
Sub bl, '0'
```

```
Shl eax, 3
```

```
Add eax, ebx
```

```
Jmp L
```

```
L1:
```

```
Outword eax
```

```
Exit
```

```
End Start
```

4.2.22. Написать полную программу решения задачи. Дана непустая последовательность символов из набора ['0'...'3'] – запись числа в 4-ой системе, за последовательностью следует пробел. Ввести это число и напечатать его в десятичной системе. Считать, что запись числа корректная и что само число умещается в двойное слово. Операцию mul не использовать.

Include console.inc

.code

Start:

Xor eax, eax

Xor ebx, ebx

L:

Inchar bl, ' '

Je L1

Sub bl, '0'

Shl eax, 2

Add eax, ebx

Jmp L

L1:

Outword eax

Exit

End Start

4.3.22. Alp DB 'abcdefghijklmnopqrstuvwyz' , 0. Написать полную программу решения задачи. Ввести число k ($1 \leq k \leq 26$) . Используя макрокоманду OUTSTR, напечатать алфавит Alp без k-той буквы в две строки: в первой строке символы, идущие в алфавите после k-той буквы, во второй – буквы перед k-той

inint eax

lea ebx, Alp[eax]

dec eax

outstr ebx

newline

mov Alp[eax], 0

outstr offset Alp

newline

4.4. Написать полную программу решения задачи. Ввести (по intint) число k ($1 \leq k \leq 30$). Используя макрокоманды outstr и newline, напечатать картинку из k строк: в первой строке напечатать k символов 'З', во второй – k-1 символ 'З', и т.д., в k-той строке – один символ 'З'. Справа дан пример для k=4.

Ограничение: цикл должен содержать не более 4-х команд, включая loop; вложенные циклы не использовать.

.data

x db '33333333333333333333333333333333',0

.code

Start:

inint ecx

lea ebx, x

mov x[ecx], 0

L:outstr ebx

newline

mov x[ecx-1], 0

Loop L

4.5. Написать полную программу решения задачи. Ввести (по intint) число k ($1 \leq k \leq 20$). Используя макрокоманды outstr и newline, напечатать картинку из k строк, каждая строка состоит из k символов: в начале строки идут минусы, затем – звёздочки. Справка дан пример для $k=4$. Программа должна содержать не более одного цикла.

```
include console.inc
```

```
.data
```

```
alp_1 db '-----', 0
```

```
alp_2 db 20 dup (?), 0
```

```
.code
```

```
Start:
```

```
inint eax
```

```
mov alp_1[eax], 0
```

```
mov ecx, eax
```

```
lea edx, alp_2
```

```
L:
```

```
call Print
```

```
mov alp_1 [ecx-1], 0
```

```
mov byte ptr [edx], '*'
```

```
inc edx
```

```
loop L
```

```
Fin:
```

```
exit
```

```
Print proc
```

```
;push ecx
```

```
lea ebx, alp_1
```

```
outstr ebx
```

```
lea ebx, alp_2
```

```
outstr ebx
```

```
newline
```

```
;pop ecx
```

```
ret
```

```
Print endp
```

```
end Start
```

4.6. Написать полную программу решения задачи. Ввести (по `intint`) число k ($1 \leq k \leq 21$). Используя макрокоманды `outstr` и `newline`, напечатать фразу «Ехал Грека через реку», начиная с k -го символа. Например, для $k=2$ нужно напечатать «хал Грека через реку». Если x не принадлежит указанному диапазону, ничего не печатать.

```
.data
```

```
    x db 'Ехал Грека через реку',0
```

```
.code
```

```
Start:
```

```
    inint ecx
```

```
    cmp ecx, 21
```

```
    ja FIN
```

```
    cmp ecx, 1
```

```
    jb FIN
```

```
    lea ecx, x[ecx-1]
```

```
    ostr ecx
```

```
    newline
```

```
FIN:
```

```
exit
```

4.7. Написать полную программу решения следующей задачи. Дан текст не длиннее 100 символов, за текстом следует точка. Ввести этот текст и напечатать его (без точки), используя макрокоманду OUTSTR. Если текст длиннее 100 символов, то выдать диагностику.

```
include console.inc
```

```
.data
```

```
    x db 100 dup(?), 0
```

```
    ermes db 'Text is larger then 100 symbols', 0
```

```
.code
```

```
Start:
```

```
    xor ecx, ecx
```

```
L: inchar al
```

```
    cmp al, '.'
```

```
    je CNT
```

```
    cmp ecx, 100
```

```
    je ERR
```

```
    mov x[ecx], al
```

```
    inc ecx
```

```
    jmp L
```

```
CNT: mov x[ecx], 0
```

```
    outstrln offset x
```

```
    jmp FIN
```

```
ERR: outstrln offset ermes
```

```
FIN: exit
```

```
end Start
```
