

??????

Задача 1.

Вычислить выражение: $(a + b) * c - d$, где $a=5$, $b=3$, $c=4$, $d=8$. Результат вывести в консоль.

```
global main
extern printf
section .data
    cura dd 5
    curb dd 3
    curc dd 4
    curd dd 8
    mymsg db "%d",10,0
section .text
main:
    mov eax, 0
    add eax, [cura]
    add eax, [curb]
    mov edx, [curc]
    mul dword [curc]; in ax (a+b)*c
    sub eax, [curd]
    mov rsi, rax ; print results
    mov rax, 0
    mov rdi, mymsg
    call printf
    ret
```

Проверяет: Работу с регистрами, базовые арифметические операции.

Задача 2.

Перевернуть строку "Hello!" и вывести. Не использовать внешние функции кроме системных вызовов.

```
global _start
section .data
    msg db "abcdefg",10,0
    msg_full_len equ $-msg
    msg_half_len equ (msg_full_len - 2)/2
section .text
```

```

_start:
; mirroring string
mov rcx, msg_half_len
mov r8, msg_full_len - 3
mov r9, 0
mov rsi, msg
loop:
mov bl, [rsi + r8]
mov dl, [rsi + r9]
mov [rsi + r9], bl
mov [rsi + r8], dl
dec r8
inc r9
dec rcx
jnz loop
; printing string
mov rax, 1
mov rdi, 1
mov rsi, msg
mov rdx, msg_full_len-1
syscall
mov rax, 60
mov rdi, 0
syscall

```

Проверяет: Работу с памятью, циклами, обработку строк.

Задача 3.

В массиве чисел [7, 2, 9, 1, 5] найти максимальный элемент и вывести его.

```

global main
extern printf

section .data
    nums db 7, 2, 9, 1, 5
    nums_len equ $-nums
    infostr db "Max number: %d",10,0

section .text
main:
    mov rcx, nums_len
    mov rax, 0
mainloop:

```

```

cmp byte [nums + rcx-1], al
ja new_max
dec rcx
jnz mainloop
jmp progend
new_max:
mov al, [nums + rcx -1]
dec rcx
jnz mainloop
progend:
mov rsi, rax
mov rax, 0
mov rdi, infostr
call printf
mov rax, 60
syscall

```

Проверяет: Работу с массивами, условные переходы.

Задача 4.

Реализуйте рекурсивную функцию вычисления факториала для n=5

```

extern printf

section .data
msg db "Factorial: %d",10,0
fact equ 5

section .text
global main

main:
mov rcx, fact
mov rax, 1
call factorial

mov rsi, rax
mov rax, 0
mov rdi, msg
call printf
mov rax, 60
xor rdi, rdi
syscall

```

```
factorial:  
    mul rcx  
    dec rcx  
    cmp rcx, 1  
    jnz factorial  
    ret
```

Проверяет: Понимание стека, рекурсии, соглашений о вызовах.

Задача 5.

Revision #11

Created 8 November 2025 05:54:43 by Admin

Updated 11 November 2025 06:56:57 by Admin