
PROGRAMOWANIE NA POZIOMIE ASEMBLERA

ZADANIA

Z1. Uzupełnij poniższe działania:

ADE + **4D5** =

9B6 - **399** =

Z2. Jak reprezentowane są liczby 3D4 i -3D4 w komputerze w rejestrze 16-bitowym?

.....

Z3. Jaka jest wartość bezwzględna liczby reprezentowanej w rejestrze 16-bitowym jako FFD4?

.....

Z4. Dany jest następujący program w assemblerze:

```
mov ax,3
mov bx,5
add ax,bx
neg bx
add bx,ax
sub ax,bx
int 3
```

Jaka będzie wartość rejestrów *ax* i *bx* po wykonaniu programu?

AX = BX =

Z5. Dany jest następujący program w assemblerze:

```
mov ax,9
et1: cmp ax,4
     jle et2
     sub ax,2
     jmp et1
et2: int 03
```

Jaka będzie wartość rejestru *ax* po wykonaniu programu?

AX =

Z6. Jaka (heksadecymalnie) wartość znajdzie się w rejestrze AX w wyniku wykonania poniższego programu:

```
mov dx, 2
add dx, dx
mov cx, 3
mov ax, 2
mul cx
add ax, dx
int 3
```

AX =

Z7. Jaka (heksadecymalnie) wartość znajdzie się w rejestrze AX w wyniku wykonania poniższego programu:

```
mov dx, 3
mov ax, 1
lup: add ax, ax
sub dx, 1
jg lup
mov bx, 5
div bx
add ax, dx
int 3
```

AX =

Z8. Jaki będzie wynik działania poniższej sekwencji instrukcji?

```
_start:
mov ax, 13
mov bx, ax
sub dx, dx
sub cx, cx
loop:
cmp dx, bx
je fini
push bx
mov bx, 16
mul bx
pop bx
add cx, 1
jmp loop
fini:
printReg cx
return0
```

Makra printReg i return0 mają standardowe znaczenie (printReg wyświetla zawartość wskazanego rejestru, a return0 powoduje zakończenie wykonywania programu).

CX =

Z9. Dany jest następujący program w języku NASM:

```
%include AuxMacros.asm
section .text
global _start
%define y 3
%define z 7
%define x 5
_start:
    mov ax, y
    mov bx, z
    mov cx, x
    push cx
    push dx
    call f
    printReg ax
    return0
f: mov si, ax
   add ax, ax
   call g
   mul cx
   add ax, si
   ret
g: mul cx
   add ax, bx
   pop dx
   pop cx
   ret
section .data
HexDig db '0', '1', '2', '3'
       db '4', '5', '6', '7'
       db '8', '9', 'A', 'B'
       db 'C', 'D', 'E', 'F'
msg db '12 = 0000', 0xa
len equ $ - msg
```

Jaki będzie wynik działania tego programu? Weź pod uwagę, że instrukcje CALL i RET manipulują adresami 32-bitowymi.

AX =

Z10. Dany jest poniższy program. Aby się skompilował, w miejsce trzech kropek trzeba wpisać liczbę.

```
%include AuxMacros.asm
section .text
global _start
%define n 3
%define x ...
_start:
    mov bx, 1
    mov cx, n
    mov ax, bx
loop: cmp cx, 0
    je fini
    mov dx, x
    mul dx
    call a
    add ax, bx
    sub cx, 1
    jmp loop
fini: printReg ax
    return0
a: neg bx
    ret
section .data
HexDig db '0', '1', '2', '3'
        db '4', '5', '6', '7'
        db '8', '9', 'A', 'B'
        db 'C', 'D', 'E', 'F'
msg db '12 = 0000', 0xa
len equ $ - msg
```

Jaka ma być wartość x , aby po wykonaniu programu było $ax = FFFC$?

Odp =

Z11. Pewna maszyna Turinga jest zdefiniowana następującą tablicą (funkcją) przejść:

	0	1	b
S_0	$\langle 0, R, S_0 \rangle$	$\langle 1, R, S_0 \rangle$	$\langle 0, R, S_1 \rangle$
S_1	-	-	$\langle 0, R, S_2 \rangle$
S_2	-	-	$\langle 1, R, S_3 \rangle$

S_0 jest stanem początkowym, a S_3 końcowym. Na początku obliczeń na taśmie znajduje się liczba naturalna x zapisana w systemie binarnym, po której następuje nieskończona sekwencja symboli b (b jak *blank*, czyli pusty). Niech y oznacza wynik działania tej maszyny. Opisz działanie tej maszyny za pomocą operacji arytmetycznych (+, −, •, /).

Odp.: $y =$ _____

Z12. Na taśmie maszyny Turinga jest zapisana – binarnie, w uzupełnieniu do 2 – liczba całkowita poprzedzona literą x . Liczba bitów nie jest z góry określona, ale każda liczba nieujemna zaczyna się od 0. Na przykład $x0111$ oznacza $+7$, a $x111$ to -1 . Należy napisać program (tablicę przejść) dla tej maszyny Turinga tak, aby zmieniała znak liczby umieszczonej na taśmie. Liczba cyfr binarnych ma pozostać bez zmian. Po zakończeniu obliczeń głowica ma być na najmniej znaczącej cyfrze. Na przykład dla $x111$ powinniśmy otrzymać $x001$ i głowica powinna być na cyfrze 1 , a dla $x0111$ wynikiem powinno być $x1001$.

Uzupełnij poniższą tablicę przejść. Stany oznaczono w niej literami alfabetu greckiego. Stanem początkowym jest α , a stanem końcowym Ω .

	x	0	1	b
α	$\langle x, R, \beta \rangle$	-	-	-
β	-	$\langle 1, R, \beta \rangle$	$\langle 0, R, \beta \rangle$	$\langle b, L, \gamma \rangle$
γ				-
δ	-	$\langle 0, R, \delta \rangle$	$\langle 1, R, \delta \rangle$	$\langle b, L, \Omega \rangle$