

## Sprawozdanie laboratoria 1

Aby wykonać obecne sprawozdanie w pierwszej kolejności musimy pobrać „symulator MARS” ze strony <http://courses.missouristate.edu/KenVollmar/MARS/> i zainstalować w systemie. W razie problemów z java należy pobrać je ze stron <https://www.oracle.com/java/technologies/downloads/#jdk20-windows> i również zainstalować.

The screenshot shows two web pages side-by-side. On the left is the MARS (MIPS Assembler and Runtime Simulator) website, which includes a navigation menu with links like Home, Features, Download, License, Papers, Help & Info, and Contact Us. It also features a '100% FREE' badge from Softpedia and a download link for MARS 4.5 software. On the right is the Java Oracle website, showing the Java logo and a section titled 'Jak chcesz otworzyć ten plik?' (How do you want to open this file?) with a button for 'Nadal używaj tej aplikacji' (Still use this application) and a link to 'Java(TM) Platform SE binary'. Below this is a section for 'Inne opcje' (Other options) with a link to 'Mars4\_5.jar'.

Po instalacji i uruchomieniu przestępuje do **zadania nr. 1**, które polega na pobraniu od użytkownika trzech liczb i wykonania operacji  $7*(a + b) + 2*b + c$  i wyświetli wynik. Zawartość pliku zad1.asm

```
.data
liczba_a: .asciiz "wprowadz a: "
liczba_b: .asciiz "wprowadz b: "
liczba_c: .asciiz "wprowadz c: "
wynik: .asciiz "wynik: "

.text
a:
    # syscall 4, wyswietlanie tekstu
    li $v0, 4
    la $a0, liczba_a
    syscall

    # syscall 5, pobieranie tekstu od uzytkownika
    li $v0, 5
    syscall
    move $t0, $v0

b:
    li $v0, 4
    la $a0, liczba_b
    syscall

    li $v0, 5
    syscall
    move $t1, $v0

c:
    li $v0, 4
    la $a0, liczba_c
    syscall

    li $v0, 5
    syscall
    move $t2, $v0

operacje:
    # dodawanie a oraz b
    add $s0, $t0, $t1

    # mnozenie 7 oraz (a + b)
    li $v0, 7
```

```

mult $s0, $v0
mflo $s1

# mnozenie 2 oraz b
li $v0, 2
mult $t1, $v0
mflo $s2

# dodawanie 7 * (a + b) oraz 2 * b
add $s3, $s1, $s2

# 7 * (a + b) + 2 * b + c
add $s4, $s3, $t2

wyswietlanie_wyniku:
li $v0, 4
la $a0, wynik
syscall

# syscall 1, wyswietlanie liczby
li $v0, 1
move $a0, $s4
syscall

koniec_programu:
# syscall 10, terminacja programu
li $v0, 10
syscall

```

Mars MessagesRun I/ORegistersCoproc 1Coproc 0

wprowadz a: 5  
wprowadz b: 5  
wprowadz c: 2  
wynik: 82  
-- program is finished running --

Clear

Labels

Label	Address ▲
zad1.asm	
a	0x00400000
b	0x0040001c
c	0x00400038
operacje	0x00400054
wyswietlanie_wyn...	0x00400078
koniec_programu	0x00400094
liczba_a	0x10010000
liczba_b	0x1001000d
liczba_c	0x1001001a
wynik	0x10010027

Name	Number	Value
\$zero	0	0x00000000
\$at	1	0x10010000
\$v0	2	0x0000000a
\$v1	3	0x00000000
\$a0	4	0x00000052
\$a1	5	0x00000000
\$a2	6	0x00000000
\$a3	7	0x00000000
\$t0	8	0x00000005
\$t1	9	0x00000005
\$t2	10	0x00000002
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x0000000a
\$s1	17	0x00000046
\$s2	18	0x0000000a
\$s3	19	0x00000050
\$s4	20	0x00000052
\$s5	21	0x00000000
\$s6	22	0x00000000
\$s7	23	0x00000000
\$t8	24	0x00000000
\$t9	25	0x00000000
\$k0	26	0x00000000
\$k1	27	0x00000000
\$gp	28	0x10008000
\$sp	29	0x7ffffeffc
\$fp	30	0x00000000
\$ra	31	0x00000000
pc		0x0040009c
hi		0x00000000
lo		0x0000000a

Zmiany w rejestrach:

- 1
- 2
- 4
- 8
- 9
- 10
- 16
- 17
- 18
- 19
- 20

Zmiany w rejestrach:

- 1
- 2
- 4
- 8
- 9
- 10
- 16
- 17
- 18
- 19
- 20
- 29 pozostał bez zmian

**Zadanie 2** polegało na utworzeniu pętli w celu dodania dwóch tablic z 4 zadeklarowanymi wartościami. Zawartość pliku zad2.asm:

```
.data
lista_1: .word 1, 3, 4, 8
lista_2: .word 7, 5, 4, 0
nowa_lista: .word, 0, 0, 0, 0
wynik: .asciiz "wynik: "
spacja: .asciiz " "

.text
inicjalizacja:
# listy
la $t0, nowa_lista
la $t1, lista_1
la $t2, lista_2

# licznik
li $s0, 0

wyswietlanie_wyniku:
li $v0, 4
la $a0, wynik
syscall

petla:
# dodawanie elementow
lw $t3, ($t1)
lw $t4, ($t2)
add $t5, $t3, $t4
sw $t5, ($t0)

# wyswietlanie cyfry
li $v0, 1
move $a0, $t5
syscall

# wyswietlanie odstepu
li $v0, 4
la $a0, spacja
syscall

# powiekszenie licznika
addi, $t0, $t0, 4
addi $t1, $t1, 4
addi $t2, $t2, 4
addi $s0, $s0, 1

# powrot po warunku
blt $s0, 4, petla

koniec_programu:
li $v0, 10
syscall
```

**Mars Messages**   **Run I/O**

wynik: 8 8 8 8  
-- program is finished running --

Clear

**Labels**

Label	Address ▲
<b>zad2.asm</b>	
inicjalizacja	0x00400000
wyswietlanie_wyn...	0x0040001c
petla	0x0040002c
koniec_programu	0x00400070
lista_1	0x10010000
lista_2	0x10010010
nowa_lista	0x10010020
wynik	0x10010030

☒ Data   ☒ Text

Dla tablic 1, 3, 4, 8  
7, 5, 4, 0

Zmiany w rejestrach: 2, 4, 8, 9, 10, 11, 13, 16

- 29 bez zmian

**Zadanie 3** polegało na opracowaniu kodu kalkulatora, który umożliwi użytkownikowi wybór typu operacji (dodawanie lub mnożenie) a następnie pobierze od użytkownika dwie liczby, wykona na nich wybraną operację i wyświetli wynik. Zawartość pliku zad3a.asm:

```
.data
    operacja: .asciiz "wpisz 1 dla dodawania, wpisz 2 dla odejmowania: "
    liczba_a: .asciiz "pierwsza wartosc: "
    liczba_b: .asciiz "druga wartosc: "
    wynik: .asciiz "wynik: "

.text
    typu_operacji:
        li $v0, 4
        la $a0, operacja
        syscall

        li $v0, 5
        syscall
        move $t0, $v0

    sprawdzenie_poprawnosci:
        beq $t0, 1, wybor_pierwszej_wartosci
        beq $t0, 2, wybor_pierwszej_wartosci

    koniec_programu:
        li $v0, 10
        syscall

    wybor_pierwszej_wartosci:
        li $v0, 4
        la $a0, liczba_a
        syscall

        li $v0, 5
        syscall
        move $t1, $v0

    wybor_drugiej_wartosci:
        li $v0, 4
        la $a0, liczba_b
        syscall

        li $v0, 5
        syscall
        move $t2, $v0

    skok_do_wlasciwej_operacji:
        beq $t0, 1, dodawanie
        beq $t0, 2, odejmowanie

    dodawanie:
        add $s0, $t1, $t2

        # ominiecie odejmowania
        j wyswietlanie_wyniku

    odejmowanie:
        sub $s0, $t1, $t2

    wyswietlanie_wyniku:
        li $v0, 4
        la $a0, wynik
        syscall

        li $v0, 1
        move $a0, $s0
        syscall

        # koniec
        j koniec_programu
```

Mars Messages

Run I/O

Clear

wpisz 1 dla dodawania, wpisz 2 dla odejmowania: \*\*\*\* user input : 2  
 pierwsza wartosc: \*\*\*\* user input : 2023  
 druga wartosc: \*\*\*\* user input : 1984  
 wynik: 39  
 -- program is finished running --

Label	Address ▲				
zad3a.asm					
typu_operacji	0x00400000		\$zero	0	0x00000000
sprawdzenie_pop...	0x0040001c		\$at	1	0x10010000
koniec_programu	0x0040002c		\$v0	2	0x0000000a
wybor_pierwszej_...	0x00400034		\$v1	3	0x00000000
wybor_drugiej_w...	0x00400050		\$a0	4	0x00000027
skok_do_wlasciw...	0x0040006c		\$a1	5	0x00000000
dodawanie	0x0040007c		\$a2	6	0x00000000
odejmowanie	0x00400084		\$a3	7	0x00000000
wyswietlanie_wyn...	0x00400088		\$t0	8	0x00000002
operacja	0x10010000		\$t1	9	0x000007e7
liczba_a	0x10010031		\$t2	10	0x000007c0
liczba_b	0x10010044		\$t3	11	0x00000000
wynik	0x10010054		\$t4	12	0x00000000
			\$t5	13	0x00000000
			\$t6	14	0x00000000
			\$t7	15	0x00000000
			\$s0	16	0x00000027
			\$s1	17	0x00000000
			\$s2	18	0x00000000
			\$s3	19	0x00000000
			\$s4	20	0x00000000
			\$s5	21	0x00000000
			\$s6	22	0x00000000
			\$s7	23	0x00000000
			\$t8	24	0x00000000
			\$t9	25	0x00000000
			\$k0	26	0x00000000
			\$k1	27	0x00000000
			\$gp	28	0x10008000
			\$sp	29	0x7ffffc
			\$fp	30	0x00000000
			\$ra	31	0x00000000
			pc		0x00400034
			hi		0x00000000
			lo		0x00000000

Zmiany w rejestrach: 1, 2, 4, 8, 9, 10, 16, 28

- 29 bez zmian