

## Зертханалық жұмыс №10

**Пән:** Компьютер архитектурасы және операциялар келісімділігі

**Тақырыбы:** MARS MIPS массив элементтерін жады ұяшықтарына кәштеу

**Мақсаты:**

- Массивтерді өңдеудің негізгі алгоритмдерін MIPS тілінде қолдану.
- Массив элементтерімен жұмыс істеу дағдыларын жетілдіру.
- Іздеу және сұрыптау алгоритмдерін түсіну.

### Қысқаша теория

Массив – бір типті деректердің реттелген жиынтығы. MIPS-те массивтер `.data` бөлімінде анықталады, әр элемент жадта бір-бірінің артында орналасады. Массив элементіне қол жеткізу үшін оның базалық адресі мен индексін пайдалану керек.

**Массив элементіне қол жеткізу мысалы:**

```
asm
la $s0, array      # Массивтің базалық адресі
li $t0, 2           # Индекс (i = 2)
sll $t1, $t0, 2     # i * 4 (байттарға)
add $t1, $s0, $t1   # Элементтің адресі
lw $t2, 0($t1)      # array[2] мәнін оқу
```

**Массивті анықтау мысалы:**

```
asm
.data
array: .word 2, 4, 6, 8, 10 # 5 элементті бүтін сандар массиві
size:  .word 5              # Массив өлшемі
```

## Зертханалық жұмыс №10 тапсырмалары:

1. **Массив қосындысы:** 5 элементті массивтің барлық элементтерінің қосындысын есептеп шығарыңыз.

```
asm
```

```
.data
arr1: .word 3, 5, 7, 2, 9
size1: .word 5
```

2. **Максимумды табу:** Массивтегі ең үлкен элементті тауып шығарыңыз.

```
asm
```

```
.data
arr2: .word 12, 45, 23, 67, 34
size2: .word 5
```

3. **Минимумды табу:** Массивтегі ең кіші элементті тауып шығарыңыз.

```
asm
```

```
.data
arr3: .word 56, 23, 78, 12, 45
size3: .word 5
```

4. **Элемент іздеу:** Массивте берілген санды іздеп, оның индекcін шығарыңыз. Егер табылмаса, -1 шығарыңыз.

```
asm
```

```
.data
arr4: .word 5, 8, 3, 9, 2
size4: .word 5
search_val: .word 9 # Ізделетін мән
```

5. **Орташа мән:** Массив элементтерінің арифметикалық ортасын есептеңіз (бүтін сан ретінде).

```
asm
```

```
.data
arr5: .word 10, 20, 30, 40, 50
size5: .word 5
```

## Мысал шешім (4-тапсырма үшін):

```
asm
.data
arr4: .word 5, 8, 3, 9, 2
size4: .word 5
search_val: .word 9
found_msg: .ascii "Индекс: "
not_found_msg: .ascii "Табылмады"

.text
main:
    # Инициализация
    la $s0, arr4          # Массив адресі
    lw $s1, size4          # Массив өлшемі
    lw $s2, search_val     # Ізделетін мән
    li $t0, -1             # Табылған индекс (бастапқыда -1)
    li $t1, 0              # Қазіргі индекс (i = 0)

search_loop:
    bge $t1, $s1, check_result # Цикл шарты

    # array[i] мәнін алу
    sll $t2, $t1, 2         # i * 4
    add $t2, $s0, $t2       # array[i] адресі
    lw $t3, 0($t2)          # array[i] мәні

    # Санды тексеру
    beq $t3, $s2, found     # Егер табылса

    addi $t1, $t1, 1        # i++
    j search_loop

found:
    move $t0, $t1           # Индексі сақтау
    j check_result

check_result:
    # Нәтижені шығару
    bltz $t0, not_found     # Егер индекс -1 болса

    # Табылған жағдай
    la $a0, found_msg
    li $v0, 4
```

```
syscall
move $a0, $t0
li $v0, 1
syscall
j exit
```

```
not_found:
    # Табылмаған жағдай
    la $a0, not_found_msg
    li $v0, 4
    syscall
```

```
exit:
    # Бағдарламаны аяқтау
    li $v0, 10
    syscall
```

### **Әр тапсырманы орындау алгоритмі:**

1. Массивтің базалық адресін жүктеу
2. Массив өлшемін жүктеу
3. Цикл арқылы барлық элементтерді өңдеу
4. Қажетті операцияны орындау (қосу, салыстыру, іздеу)
5. Нәтижені экранға шығару