

コンピュータアーキテクチャ演習 1 回

課題1-1

ソースコード

```
# データセグメント
        .data
A:      .word 19
B:      .word 75
C:      .word 10
S:      .word 0

# テキストセグメント
        .text
main:
    # メモリからレジスタにデータをロード
    lw   $8, A
    lw   $9, B
    lw   $10, C
    # 演算
    add  $11, $8, $9
    sub  $11, $11, $10
    ori  $11, $11, 3
    # レジスタからメモリにデータを保存
    sw   $11, S
exit:
    j    exit
```

結果

$$S = d9_{(16)}$$

課題1-2

ソースコード

```
        .data
N:      .word 10
A:      .word 8
        .word 4
        .word 7
        .word 12
```

```

        .word    13
        .word    19
        .word    23
        .word    43
        .word    56
        .word    32
S:      .word    0

        .text
main:
    or    $8, $0, $0 # i = 0
    la    $9, A # 配列の先頭のアドレスを読み取る
    or    $10, $0, $0 # 合計値の初期化
    lw    $11, N
loop:
    beq   $8, $11, loopend # i == N になったらループを抜ける
    lw    $12, 0($9) # 配列のi番目をロード
    add   $10, $10, $12 # sum += A[i]
    addi  $9, $9, 4 # addr += 4
    addi  $8, $8, 1 # i++
    j     loop # ループの先頭に戻る

loopend:
    sw    $10, S # レジスタからメモリに保存

exit:
    j     exit

```

結果

$$S = 57_{(16)}$$

課題1-3

ソースコード

```

        .data
N:      .word    10
A:      .word    8
        .word    4
        .word    7
        .word    12
        .word    13
        .word    19
        .word    23
        .word    43
        .word    56

```

```

        .word    32
B:      .space   40

        .text
main:
    or    $8, $0, $0 # i = 0
    la    $9, A      # arr A
    la    $10, B     # arr B
    lw    $11, N      # N

loop:
    beq   $8, $11, loopend # i == N ならループを抜ける
    lw    $12, 0($9) # Aのi番目をメモリからレジスタにロード
    sw    $12, 0($10) # Bのi番目にレジスタから保存
    addi  $9, 4 # Aのアドレスを進める
    addi  $10, 4 # Bのアドレスを進める
    addi  $8, 1 # i++
    j     loop # ループの先頭に戻る

loopend:

exit:    j exit

```

結果

Data Segments				
DATA				
[0x00005000]	0x0000000a	0x00000008	0x00000004	0x00000007
[0x00005010]	0x0000000c	0x0000000d	0x00000013	0x00000017
[0x00005020]	0x0000002b	0x00000038	0x00000020	0x00000008
[0x00005030]	0x00000004	0x00000007	0x0000000c	0x0000000d
[0x00005040]	0x00000013	0x00000017	0x0000002b	0x00000038
[0x00005050]	0x00000020	0x00000000	0x00000000	0x00000000
[0x00005060]... [0x00025000]	0x00000000			

課題1-4

ソースコード

```

        .data
N:      .word    10
A:      .word    8
        .word    4
        .word    7
        .word    12
        .word    13
        .word    19
        .word    23
        .word    43
        .word    56
        .word    32

```

```

.text
main:
    # 初期化
    or  $8, $0, $0 # i = 0
    la  $11, A      # arr = A
    lw  $12, N      # 配列の個数を読み込み

outloop:
    slt $24, $8, $12 # i < N だったら
    beq $24, $0, loopend # loopendに移動
    or  $9, $0, $0 # j = 0
    or  $10, $11, $0 # tmp = A
    inloop:
        sub $25, $12, $8 # N-i
        addi $24, $9, 1 # j+1
        slt $24, $24, $25 # j+1 < N-i だったら
        beq $24, $0, outinc # inloop を抜けて outinc に移動

        lw $14, 0($10) # A[j]
        lw $15, 4($10) # A[j+1]

        slt $24, $15, $14 # A[j+1] < A[j] だったら
        beq $24, $0, ininc # ininc に移動
        sw $15, 0($10) # A[j] = A[j+1]
        sw $14, 4($10) # A[j+1] = A[j]

        j ininc # inincに移動

    outinc:
        addi $8, $8, 1 # i++
        addi $11, $11, 4 # 配列を 1 進める (アドレスに4足す)
        j outloop # outloop に移動

    ininc:
        addi $9, $9, 1 # j++
        addi $10, $10, 4 # tmp を 1 進める
        j inloop # inloop に移動

loopend:

exit:
    j  exit

```

結果

Data Segments					
DATA					
[0x00005000]	0x0000000a	0x00000004	0x00000007	0x00000008	
[0x00005010]	0x0000000c	0x0000000d	0x00000013	0x00000017	
[0x00005020]	0x00000020	0x0000002b	0x00000038	0x00000000	
[0x00005030] ... [0x00025000]	0x00000000				
STACK					
[0x7ffefffc]	0x00000000				