- REPORT -

Name:Đào Minh Châu

ID:24560025

1. Classwork: Answer all question in the instruction.

Question 1.1.

PC	$ra	$sp	$fp
0x00400000	0x00000000	0x7fffeffc	0x00000000
0x0040000c	0x00000004	0x7fffeffc	0x00000000
0x00400010	0x40000004	0x7fffeffc	0x00000000
0x00400014	0x40000004	0x7fffeffc	0x00000000
0x00400018	0x40000004	0x7fffeffc	0x00000000
0x0040001c	0x40000004	0x7fffeffc	0x00000000
0x00400020	0x40000004	0x7fffeffc	0x00000000
0x00400028	0x40000004	0x7fffeffc	0x00000000

Question 1.2.

The program ask the user to input a number using a procedure.

Question 1.3.

When the instruction j exit is deleted, the program doesn't end and continue to run the procedure a again which requires user to enter another number one more time.

Question 1.4.

The procedure is named “getInt”.

Question 1.5.

```
.data
prompt: .asciiz "Enter one number: "
.text
main:
  jal getInt
  add $s0, $v0, $zero
  jal showInt
  j exit
getInt:
  li $v0, 4
  la $a0, prompt
  syscall
  li $v0, 5
  syscall
  jr $ra
showInt:
  addi $a0, $s0, 1
  li $v0, 1
  syscall
exit
```

Question 2.1.

	PC	$ra	$sp	$fp
1	0x00000000	0x00000000	0x7fffeffc	0x00000000
2	0x00400004	0x00000000	0x7fffeffc	0x00000000
3	0x00400008	0x00000000	0x7fffeffc	0x00000000
4	0x0040000c	0x00000000	0x7fffeffc	0x00000000
4	0x00400010	0x00000000	0x7fffeffc	0x00000000
6	0x00400014	0x00000000	0x7fffeffc	0x00000000
7	0x00400018	0x00000000	0x7fffeffc	0x00000000
8	0x0040001c	0x00000000	0x7fffeffc	0x00000000
9	0x00400020	0x00000000	0x7fffeffc	0x00000000
10	0x00400034	0x00400024	0x7fffeffc	0x00000000
11	0x00400038	0x00400024	0x7fffeff8	0x00000000
12	0x0040003c	0x00400024	0x7fffeff8	0x00000000
13	0x00400040	0x00400024	0x7fffeff8	0x00000000
14	0x00400044	0x00400024	0x7fffeff8	0x00000000
15	0x00400048	0x00400024	0x7fffeff8	0x00000000
16	0x0040004c	0x00400024	0x7fffeff8	0x00000000
17	0x00400050	0x00400024	0x7fffeff8	0x00000000
18	0x00400054	0x00400024	0x7fffeffc	0x00000000
19	0x00400024	0x00400024	0x7fffeffc	0x00000000
20	0x00400028	0x00400024	0x7fffeffc	0x00000000

21	0x0040002c	0x00400024	0x7fffeffc	0x00000000
22	0x00400030	0x00400024	0x7fffeffc	0x00000000
23	0x00400058	0x00400024	0x7fffeffc	0x00000000

Question 2.2.

$a0: stores copy value of A

$a1: stores copy value of B

$a2: stores copy value of C

$a3: stores copy value of D

Question 2.3.

The procedure in the program is named “calculate”.

Question 2.4.

The first instruction decreases the value of stack pointer by 4(which means decrease 1 word in the memory).

The second one is to load the value in $s0 to the memory cell which is pointed by $sp.

If we remove them, the value of $s0 will be overwritten since we use the same register to store the result.

Question 2.5.

The first instruction is to load the value from the pointed cell back to $s0.

The first instruction increases the value of stack pointer by 4(which means increase 1 word in the memory).

If we remove those instructions, the original value of $s0 will not be written back to $s0 and cause data lost.

Question 2.6.

$v0 stores the temporary value of the result and to store the service numbers.

2. Exercise

2.1. Rewrite the Example 2.1 with following requirements:

- Let user input `numberA ($s0), numberB ($s1), numberC ($s2), numberD ($s3)`
- The procedure `proc_example` calculates:

  `(numberA + numberB) - (numberC + numberD)` and

  `(numberA - numberB) + (numberC - numberD)`
- The procedure `proc_example` uses `$s0` to store the result of `(numberA + numberB) - (numberC + numberD)` and `$s1` to store the result of `(numberA - numberB) + (numberC - numberD)`
- The procedure `proc_example` returns 2 values while restoring the original values of `$s0` and `$s1`. (*Hint: return an array*)
- Print the results

```
.data
input: .asciiz "Input number:"
output1: .asciiz "Result of (A+B)-(C+D):"
output2: .asciiz "Result of (A-B)+(C-D):"
a: .asciiz "$s0="
b: .asciiz "$s1="
.text
li $v0, 4
```

```
la $a0, input
syscall
li $v0, 5
syscall
addi $s0, $v0, 0
li $v0, 4
la $a0, input
syscall
li $v0, 5
syscall
addi $s1, $v0, 0
li $v0, 4
la $a0, input
syscall
li $v0, 5
syscall
addi $s2, $v0, 0
li $v0, 4
la $a0, input
syscall
li $v0, 5
syscall
addi $s3, $v0, 0
  add $s4, $s0, $zero #$s0 stores numberA
  add $s5, $s1, $zero #$s1 stores numberB
  add $s6, $s2, $zero #$s2 stores numberC
  add $s7, $s3, $zero #$s3 stores numberD

  jal proc_example
  li $v0, 4
  la $a0, output1
  syscall
  li $v0, 1
  addi $a0, $t0, 0
  syscall
  li $v0, 4
  la $a0, output2
  syscall
  li $v0, 1
  addi $a0, $t1, 0
  syscall
  j exit
proc_example:
  add $t0, $s4, $s5
  add $t1, $s6, $s7
  sub $t0, $t0, $t1

  sub $t0, $s4, $s5
  sub $t1, $s6, $s7
  add $t1, $t0, $t1
  jr $ra
exit:
```

```
li $v0, 4
la $a0, a
syscall
li $v0, 1
addi $a0, $s0, 0
syscall
li $v0, 4
la $a0, b
syscall
li $v0, 1
addi $a0, $s1, 0
syscall
```