Homework Assignment 3

(Due 2:10pm, Nov. 8, scan (or take a photo) and upload it in Canvas)

You should use the following instructions only.

- Instructions
    - ADD, SUB
    - AND, ORR, EOR
    - CMP, BGE/BLT/BGT/BLE/BEQ/BNE
    - B, BL, BX
   - MOV
    - LDR, STR

1. (50 points) Write an assembly code for the following C code (the line c=com() in the main function and the com() function).

```
int main () {
  int a, b, c, d;
  ...
  d = com (a, b, c);
  ...
}

int com (int x, int y, int z) {
  if ( x == 2 )
    return y;
  else if ( x == 4 )
    return z;
  else if ( x == 6 )
    return (y + z);

  return x*x;
}
```

a
b
c
d	← SP
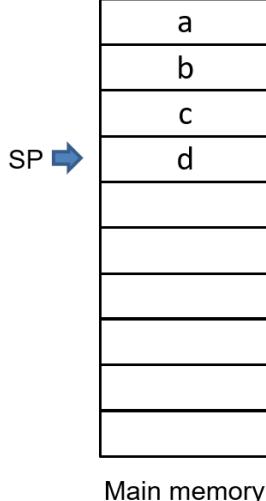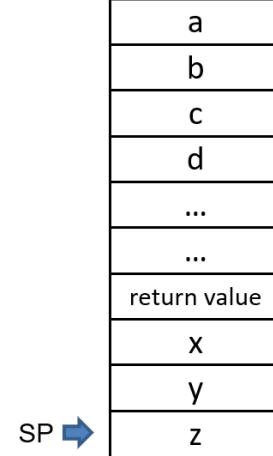
Main memory

(b)

a
b
c
d
...
...
return value
x
y
z	← SP

(c)

(a)

- In the main function, assume that R0-R12 are being used by other variables (right before the function call c=com(a,b,c)). This means that you should preserve their values if you want to use any of them.
- Use the stack memory for the function arguments and the return value (shown in (c)).
- You should implement the multiplication yourself. (Do not use MUL)

```
main:
  PUSH {R0}

  SUB SP, SP, #16
  LDR R0, [SP, #32]  // a
  STR R0, [SP, #8]
  LDR R0, [SP, #28]  // b
  STR R0, [SP, #4]
  LDR R0, [SP, #24]  // c
  STR R0, [SP]

  BL com

  LDR R0, [SP, #12]
  STR R0, [SP, #20]  // d
  ADD SP, SP, #16

  POP {R0}
```

```
com:
  PUSH {R0, R1, R2}
  LDR R0, [SP, #20]  // x
  CMP R0, #2
  BEQ ret_y
  CMP R0, #4
  BEQ ret_z
  CMP R0, #6
  BEQ ret_yz
  MOV R1, #0
  MOV R2, #0
mul:
  CMP R1, R0
  BGE ret_xx
  ADD R2, R2, R0
  ADD R1, R1, #1
  B mul
ret_y:
  LDR R0, [SP, #16]  // y
  STR R0, [SP, #24]
  B ret_com
ret_z:
  LDR R0, [SP, #12]  // z
  STR R0, [SP, #24]
  B ret_com
ret_yz:
  LDR R0, [SP, #16]  // y
  LDR R1, [SP, #12]  // z
  ADD R0, R0, R1
  STR R0, [SP, #24]
  B ret_com
ret_xx:
  STR R2, [SP, #24]
ret_com:
  POP {R0, R1, R2}
  BX LR
```

2. (50 points) Write an assembly code for the following C code (the line b=add(a) and the add()
function.

```
int main () {
  int a, b;
  ...
  b = com (a);
  ...
}

int com (int x) {
  if ( x <= 2 )
    return 5;

  return com(x-1) + com(x-2);
}
```

SP ➡

a
b

Main memory

(b)

SP ➡

a
b
...
return value
x
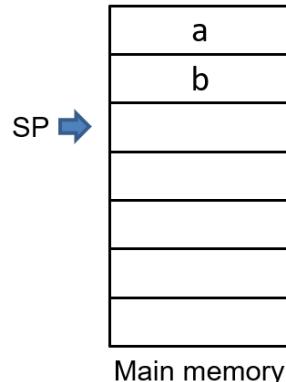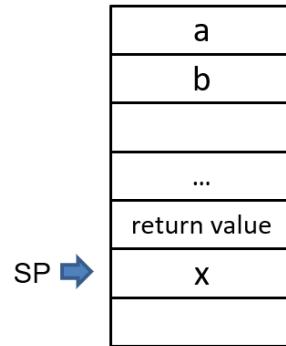
(c)

- In the main function, assume that R0-R12 are being used by other variables (right before
  the function call). This means that you should preserve their values if you want to use any
  of them.
- Use the stack memory for the function arguments and the return value (shown in (c)).

```
main:
  PUSH {R0}

  SUB SP, SP, #8
  LDR R0, [SP, #20]  // a
  STR R0, [SP]

  BL com

  LDR R0, [SP, #4]
  STR R0, [SP, #16]  // b
  ADD SP, SP, #8

  POP {R0}
```

```
com:
  PUSH {R0}
  LDR R0, [SP, #4]
  CMP R0, #2
  BGT com_ret
  MOV R0, #5
  STR R0, [SP, #8]
  POP {R0}
  BX LR
com_ret:
  SUB R0, R0, #1
  PUSH {LR}
  PUSH {R1}
  SUB SP, SP, #8
  STR R0, [SP]
  BL com
  LDR R1, [SP, #4]
  SUB R0, R0, #1
  STR R0, [SP]
  BL com
  LDR R0, [SP, #4]
  ADD R0, R0, R1
  STR R0, [SP, #24]
  ADD SP, SP, #8
  POP {R1}
  POP {LR}
  POP {R0}
  BX LR
```