

EE234

Microprocessor Systems

Final Exam

Dec. 12, 2023. (1:10pm – 4:00pm)

Instructor: Dae Hyun Kim (daehyun@eecs.wsu.edu)

Name:

WSU ID:

Problem	Points	
1	20	
2	20	
3	40	
4	20	
5	20	
6	20	
Total	140	

Problem #1 (Array, 20 points)

You can use the following instructions only in this exam.

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

Write an assembly code for the “for loop” in the following C code.

```
int* x = new int[10];
int* y = new int[10];
...
for ( int k = 0 ; k < 10 ; k++ ) {
    int a = x[k];
    x[k] = y[k];
    y[k] = a;
}
```

R15 (PC)	
R14 (LR)	0x0414
R13 (SP)	0x0408
	...
R2	...
R1	...
R0	...
	RF

0x0418	
0x0414	0x0410
0x0410	0x0408
0x040C	y
0x0408	x
0x0404	0x0404
0x0400	0x0400
	Main memory

- R0-R12 are freely available.
- You can use any of R0-R12 for “int k” and “int a” (i.e., you don’t need to use the stack for k and a).

Problem #2 (Array, 20 points)

Write an assembly code for the second “for loop” (the one in the red rectangle) in the following C code. A “long” variable occupies eight bytes.

```
long** x = new long*[20];
long** y = new long*[20];

for ( int k = 0 ; k < 20 ; k++ ) {
    x[k] = new long[20];
    y[k] = new long[20];
}

...
```

```
for ( int m = 0 ; m < 10 ; m++ ) {
    y[m][2*m] = x[2*m][m];
}
```

R15 (PC)	
R14 (LR)	0x0404
R13 (SP)	0x0404
	...
	RF

0x0418	
0x0414	0x040C
0x0410	y
0x040C	0x0414
0x0408	x
0x0404	0x0408
0x0400	0x0400

Main memory

- R0-R12 are freely available.
- You can use any of R0-R12 for “int m” (i.e., you don’t need to use the stack for “m”).

Problem #3 (Array, 40 points)

(1) How many bytes does the array “a” actually use for the array? (6 points)

```
int a[10];
```

(2) How many bytes does the array “b” actually use for the array? (6 points)

```
int* b = new int[10];
```

(3) How many bytes does the array “c” actually use for the array? (6 points)

```
int c[10][10];
```

(4) How many bytes does the array “d” actually use for the array? (6 points)

```
int** d = new int*[10];
for ( int m = 0 ; m < 10 ; m++ ) {
    d[m] = new int[10];
}
```

(5) How many bytes does the array “e” actually use for the array? (6 points)

```
long** e = new long*[10];
for ( int m = 0 ; m < 10 ; m++ ) {
    e[m] = new long[10];
}
```

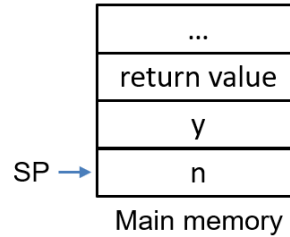
(6) We define a 5-dimensional dynamic array of long variables as follows. How many bytes does the array “g” actually use for the array? n_1, n_2, n_3, n_4, n_5 are constants and your answer should be expressed as a function of those constants. (10 points)

```
long***** g = new long****[n1];
for ( int k1 = 0 ; k1 < n1 ; k1++ ) {
    g[k1] = new long***[n2];
    for ( int k2 = 0 ; k2 < n2 ; k2++ ) {
        g[k1][k2] = new long**[n3];
        for ( int k3 = 0 ; k3 < n3 ; k3++ ) {
            g[k1][k2][k3] = new long*[n4];
            for ( int k4 = 0 ; k4 < n4 ; k4++ ) {
                g[k1][k2][k3][k4] = new long[n5];
            }
        }
    }
}
```

Problem #4 (Array, 20 points)

Write an assembly code for the com() function.

```
int com (int* y, int n) {  
    int s = 0;  
  
    for ( int k = 0 ; k < n ; k++ ) {  
        s = s + *(y+k);  
    }  
  
    return s;  
}
```

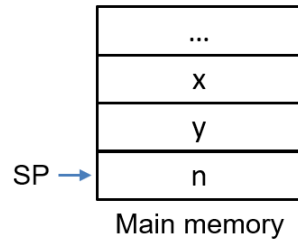


- R0-R12 are freely available.
- You can use any of R0-R12 for “int s” and “int k” in the com() function.
- Use the stack memory for the function arguments and the return value (see the figure).

Problem #5 (Array, 20 points)

Write an assembly code for the com() function. A “long” variable occupies 8 bytes.

```
void com (long* x, long* y, int n) {  
    for ( int k = 0 ; k < n ; k++ ) {  
        x[k] = y[n-k-1];  
    }  
}
```

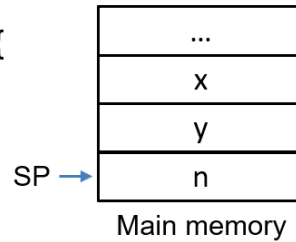


- R0-R12 are freely available.
- You can use any of R0-R12 for “int k” in the com() function.
- Use the stack memory for the function arguments and the return value (see the figure).

Problem #6 (Array, 20 points)

Write an assembly code for the com() function. A “long” variable occupies 8 bytes.

```
void com (long** x, long** y, int n) {  
    for ( int k = 0 ; k < n ; k++) {  
        for ( int s = 0 ; s < n ; s++ ) {  
            x[k][s] = y[n-s-1][n-k-1]  
        }  
    }  
}
```



- R0-R12 are freely available.
- You can use any of R0-R12 for “int k” and “int s” in the com() function.
- Use the stack memory for the function arguments and the return value (see the figure).