EXAM REVIEW

Kevin Tew

open book, open notes, open electronic notes

closed electronic tools, compilers, etc

Problem 1

Expression	Decimal Representation	7-bit Binary Representation
—	51	
—	−12	
—		`010 1010`
—		`101 0101`
`-4 << 5`		
`38 >> 3`		
TMin		
TMax		
TMin +1		
−TMax		
TMax − TMin		
TMin − TMax		

Problem 1.1

0 51 64/S 51 0000000

Problem 1.1

0 51 64/S 51 0000000

32 51 32 19 0100000

Problem 1.1

0	51	64/S	51	0000000
32	51	32	19	0100000
48	19	16	3	0110000

Problem 1.1

0	51	64/S	51	0000000
32	51	32	19	0100000
48	19	16	3	0110000
48	3	8	3	0110000

Problem 1.1

0	51	64/S	51	0000000
32	51	32	19	0100000
48	19	16	3	0110000
48	3	8	3	0110000
48	3	4	3	0110000

Problem 1.1

0	51	64/S	51	0000000
32	51	32	19	0100000
48	19	16	3	0110000
48	3	8	3	0110000
48	3	4	3	0110000
50	3	2	1	0110010

Problem 1.1

0	51	64/S	51	0000000
32	51	32	19	0100000
48	19	16	3	0110000
48	3	8	3	0110000
48	3	4	3	0110000
50	3	2	1	0110010
51	1	1	0	0110011

Problem 1

Expression	Decimal Representation	7-bit Binary Representation
—	51	0110011
—	−12	
—		010 1010
—		101 0101
`-4 << 5`		
`38 >> 3`		
TMin		
TMax		
TMin +1		
−TMax		
TMax − TMin		
TMin − TMax		

Problem 1.2

-12

Problem 1.2

-12

0001100 12

Problem 1.2

-12

0001100 12

~ 1110011 -13

Problem 1.2

-12

0001100 12

~ 1110011 -13

+1 1110100 -12

Problem 1

Expression	Decimal Representation	7-bit Binary Representation
—	51	0110011
—	−12	1110100
—		010 1010
—		101 0101
`-4 << 5`		
`38 >> 3`		
TMin		
TMax		
TMin +1		
−TMax		
TMax − TMin		
TMin − TMax		

Problem 1.3

S 0 0101010

Problem 1.3

S 0 00101010

32 32 01101010

Problem 1.3

S 0 0101010

32 32 0101010

16 32 0101010

Problem 1.3

S	0	0101010
32	32	0101010
16	32	0101010
8	40	0101010

Problem 1.3

S	0	0101010
32	32	0101010
16	32	0101010
8	40	0101010
4	40	0101010

Problem 1.3

S	0	0101010
32	32	0101010
16	32	0101010
8	40	0101010
4	40	0101010
2	42	0101010

Problem 1.3

S	0	0101010
32	32	0101010
16	32	0101010
8	40	0101010
4	40	0101010
2	42	0101010
1	42	0101010

Problem 1

Expression	Decimal Representation	7-bit Binary Representation
—	51	0110011
—	−12	1110100
—	42	010 1010
—		101 0101
`-4 << 5`		
`38 >> 3`		
TMin		
TMax		
TMin +1		
−TMax		
TMax − TMin		
TMin − TMax		

Problem 1.4

1010101 -x

Problem 1.4

1010101 -x

~ 0101010 x-1

Problem 1.4

	1010101	-x
~	0101010	x-1
+1	0101011	x

Problem 1.4

	1010101	-x
~	0101010	x-1
+1	0101011	x
	0101011	43

Problem 1.4

	1010101	-x
~	0101010	x-1
+1	0101011	x
	0101011	43
	-43	-x

Problem 1

Expression	Decimal Representation	7-bit Binary Representation
—	51	0110011
—	−12	1110100
—	42	010 1010
—	-43	101 0101
`-4 << 5`		
`38 >> 3`		
TMin		
TMax		
TMin +1		
−TMax		
TMax − TMin		
TMin − TMax		

Problem 1.5

$$-4 << 5$$

Problem 1.5

-4 << 5

0000100 4

Problem 1.5

-4 << 5

0000100 4

~ 1111011 ~4

Problem 1.5

	-4 << 5	
	0000100	4
~	1111011	~4
+1	1111100	-4

Problem 1.5

	-4 << 5	
	0000100	4
~	1111011	~4
+1	1111100	-4
<<	0000000	-4 << 5

Problem 1.5

	-4 << 5	
	0000100	4
~	1111011	~4
+1	1111100	-4
<<	0000000	-4 << 5
	0	

Problem 1

Expression	Decimal Representation	7-bit Binary Representation
—	51	0110011
—	−12	1110100
—	42	010 1010
—	-43	101 0101
`-4 << 5`	0	0000000
`38 >> 3`		
TMin		
TMax		
TMin +1		
−TMax		
TMax − TMin		
TMin − TMax		

Problem 1.6

$$38 >> 3$$

Problem 1.6

38 >> 3

0100110 38

Problem 1.6

38 >> 3

0100110 38

0000100 4

Problem 1

Expression	Decimal Representation	7-bit Binary Representation
—	51	0110011
—	−12	1110100
—	42	010 1010
—	-43	101 0101
`-4 << 5`	0	0000000
`38 >> 3`	4	0000100
TMin		
TMax		
TMin +1		
−TMax		
TMax − TMin		
TMin − TMax		

Problem 1.7

TMin $-2^{k-1},\ k = 7$

-64

1000000

Problem 1

Expression	Decimal Representation	7-bit Binary Representation
—	51	0110011
—	−12	1110100
—	42	010 1010
—	-43	101 0101
`-4 << 5`	0	0000000
`38 >> 3`	4	0000100
TMin	-64	1000000
TMax		
TMin +1		
−TMax		
TMax − TMin		
TMin − TMax		

Problem 1.8

TMax $2^{k-1} - 1,\ k = 7$

63

0111111

Problem 1

Expression	Decimal Representation	7-bit Binary Representation
—	51	0110011
—	−12	1110100
—	42	010 1010
—	-43	101 0101
`-4 << 5`	0	0000000
`38 >> 3`	4	0000100
TMin	-64	1000000
TMax	63	0111111
TMin +1		
−TMax		
TMax − TMin		
TMin − TMax		

Problem 1.9

TMin + 1

Problem 1.9

TMin + 1

TMin 1000000 -64

Problem 1.9

	TMin + 1	
TMin	1000000	-64
+1	0000001	1

Problem 1.9

	TMin + 1	
TMin	1000000	-64
+1	0000001	1
	1000001	-63

Problem 1

Expression	Decimal Representation	7-bit Binary Representation
—	51	0110011
—	−12	1110100
—	42	010 1010
—	-43	101 0101
`-4 << 5`	0	0000000
`38 >> 3`	4	0000100
TMin	-64	1000000
TMax	63	0111111
TMin +1	-63	1000001
−TMax		
TMax − TMin		
TMin − TMax		

Problem 1.10

-TMax

Problem 1.10

-TMax

TMax 0111111 63

Problem 1.10

	-TMax	
TMax	0111111	63
~	1000000	-64

Problem 1.10

	-TMax	
TMax	0111111	63
~	1000000	-64
+1	0000001	1

Problem 1.10

	-TMax	
TMax	0111111	63
~	1000000	-64
+1	0000001	1
-TMax	1000001	-63

Problem 1

Expression	Decimal Representation	7-bit Binary Representation
—	51	0110011
—	−12	1110100
—	42	010 1010
—	-43	101 0101
`-4 << 5`	0	0000000
`38 >> 3`	4	0000100
TMin	-64	1000000
TMax	63	0111111
TMin +1	-63	1000001
−TMax	-63	1000001
TMax − TMin		
TMin − TMax		

Problem 1.11

TMax-TMin

Problem 1.11

	TMax-TMin	
TMax	0111111	63

Problem 1.11

	TMax-TMin	
TMax	0111111	63
TMin	1000000	-64

Problem 1.11

	TMax-TMin	
TMax	0111111	63
TMin	1000000	-64
~TMin	0111111	63

Problem 1.11

	TMax-TMin	
TMax	0111111	63
TMin	1000000	-64
~TMin	0111111	63
+1	0000001	1

Problem 1.11

	TMax-TMin	
TMax	0111111	63
TMin	1000000	-64
~TMin	0111111	63
+1	0000001	1
-TMin	1000000	-64

Problem 1.11

	TMax-TMin	
TMax	0111111	63
TMin	1000000	-64
~TMin	0111111	63
+1	0000001	1
-TMin	1000000	-64
TMax	0111111	63

Problem 1.11

	TMax-TMin	
TMax	0111111	63
TMin	1000000	-64
~TMin	0111111	63
+1	0000001	1
-TMin	1000000	-64
TMax	0111111	63
TMax-TMin	1111111	-1

Problem 1

Expression	Decimal Representation	7-bit Binary Representation
—	51	0110011
—	−12	1110100
—	42	010 1010
—	-43	101 0101
`-4 << 5`	0	0000000
`38 >> 3`	4	0000100
TMin	-64	1000000
TMax	63	0111111
TMin +1	-63	1000001
−TMax	-63	1000001
TMax − TMin	-1	1111111
TMin − TMax		

Problem 1.12

TMin-TMax

Problem 1.12

	TMin-TMax	
TMin	1000000	-64

Problem 1.12

	TMin-TMax	
TMin	1000000	-64
TMax	0111111	63

Problem 1.12

	TMin-TMax	
TMin	1000000	-64
TMax	0111111	63
~TMax	1000000	-64

Problem 1.12

	TMin-TMax	
TMin	1000000	-64
TMax	0111111	63
~TMax	1000000	-64
+1	0000001	1

Problem 1.12

	TMin-TMax	
TMin	1000000	-64
TMax	0111111	63
~TMax	1000000	-64
+1	0000001	1
-TMax	1000001	-63

Problem 1.12

	TMin-TMax	
TMin	1000000	-64
TMax	0111111	63
~TMax	1000000	-64
+1	0000001	1
-TMax	1000001	-63
TMin	1000000	-64

Problem 1.12

	TMin-TMax	
TMin	1000000	-64
TMax	0111111	63
~TMax	1000000	-64
+1	0000001	1
-TMax	1000001	-63
TMin	1000000	-64
TMin-TMax	0000001	1

Problem 1

Expression	Decimal Representation	7-bit Binary Representation
—	51	0110011
—	−12	1110100
—	42	010 1010
—	-43	101 0101
`-4 << 5`	0	0000000
`38 >> 3`	4	0000100
TMin	-64	1000000
TMax	63	0111111
TMin +1	-63	1000001
−TMax	-63	1000001
TMax − TMin	-1	1111111
TMin − TMax	1	0000001

Problem 2

Description	Hex	M	E	Value
Negative zero		—	—	—
Positive infinity		—	—	—
	`0x3E`	—	—	—
—	`0x15`			
—				-0.25
One				1.0
Smallest denormalized > 0				
Largest normalized > 0				

Problem 2.1

Negative Zero

Problem 2.1

Negative Zero

1

Problem 2.1

Negative Zero

100

Problem 2.1

Negative Zero

100000

Problem 2.1

Negative Zero

100000

0x20

Problem 2

Description	Hex	M	E	Value
Negative zero	0x20	—	—	—
Positive infinity		—	—	—
	0x3E	—	—	—
—	0x15			
—				-0.25
One				1.0
Smallest denormalized > 0				
Largest normalized > 0				

Problem 2.2

Positive Infinity

Problem 2.2

Positive Infinity

0

Problem 2.2

Positive Infinity

011

Problem 2.2

Positive Infinity

011000

Problem 2.2

Positive Infinity

011000

0x18

Problem 2

Description	Hex	M	E	Value
Negative zero	0x20	—	—	—
Positive infinity	0x18	—	—	—
	0x3E	—	—	—
—	0x15			
—				-0.25
One				1.0
Smallest denormalized > 0				
Largest normalized > 0				

Problem 2.3

0x3E

Problem 2.3

0x3E

1

Problem 2.3

0x3E

111

Problem 2.3

0x3E

111110

Problem 2.3

0x3E

111110

QNaN

Problem 2

Description	Hex	M	E	Value
Negative zero	0x20	—	—	—
Positive infinity	0x18	—	—	—
QNaN	0x3E	—	—	—
—	0x15			
—				-0.25
One				1.0
Smallest denormalized > 0				
Largest normalized > 0				

Problem 2.4

0x15

Problem 2.4

0x15

0

Problem 2.4

0x15

010

Problem 2.4

0x15

010101

Problem 2.4

0x15

010101

Bias is 1 E is 1

Problem 2.4

0x15

010101

Bias is 1 E is 1

$$1.0 + \frac{1}{2} + \frac{1}{8} = 1.625$$

Problem 2.4

0x15

010101

Bias is 1 E is 1

$$1.0 + \frac{1}{2} + \frac{1}{8} = 1.625$$

$$1 * 2^1 * 1.625 = 3.25$$

Problem 2

Description	Hex	M	E	Value
Negative zero	0x20	—	—	—
Positive infinity	0x18	—	—	—
QNaN	0x3E	—	—	—
—	0x15	1.625	1	3.25
—				-0.25
One				1.0
Smallest denormalized > 0				
Largest normalized > 0				

Problem 2.5

-0.25

Problem 2.5

-0.25

1

Problem 2.5

-0.25

100

Problem 2.5

-0.25

100010

Problem 2.5

-0.25

100010

00 exp bits means denormalized

Problem 2.5

-0.25

100010

00 exp bits means denormalized

$$0.0 + \frac{1}{4} = 0.25$$

Problem 2.5

-0.25

100010

00 exp bits means denormalized

$$0.0 + \frac{1}{4} = 0.25$$

$$-1 * 2^0 * 0.25 = -0.25$$

Problem 2

Description	Hex	M	E	Value
Negative zero	0x20	—	—	—
Positive infinity	0x18	—	—	—
QNaN	0x3E	—	—	—
—	0x15	1.625	1	3.25
—	0x22	0.25	0	-0.25
One				1.0
Smallest denormalized > 0				
Largest normalized > 0				

Problem 2.6

1.0

Problem 2.6

1.0

0

Problem 2.6

1.0

001

Problem 2.6

1.0

001000

Problem 2.6

1.0

001000

Bias is 1 E is 0

Problem 2.6

1.0

001000

Bias is 1 E is 0

1.0

Problem 2.6

1.0

001000

Bias is 1 E is 0

$$1.0$$

$$1 * 2^0 * 1.0 = 1.0$$

Problem 2

Description	Hex	M	E	Value
Negative zero	0x20	—	—	—
Positive infinity	0x18	—	—	—
QNaN	0x3E	—	—	—
—	0x15	1.625	1	3.25
—	0x22	0.25	0	-0.25
One	0x08	1.0	0	1.0
Smallest denormalized > 0				
Largest normalized > 0				

Problem 2.7

Smallest denormalized > 0

Problem 2.7

Smallest denormalized > 0

0

Problem 2.7

Smallest denormalized > 0

000

Problem 2.7

Smallest denormalized > 0

000001

Problem 2.7

Smallest denormalized > 0

000001

00 exp bits means denormalized

Problem 2.7

Smallest denormalized > 0

000001

00 exp bits means denormalized

$$0.0 + \frac{1}{8} = 0.125$$

Problem 2.7

Smallest denormalized > 0

000001

00 exp bits means denormalized

$$0.0 + \frac{1}{8} = 0.125$$

$$1 * 2^0 * 0.125 = 0.125$$

Problem 2

Description	Hex	M	E	Value
Negative zero	0x20	—	—	—
Positive infinity	0x18	—	—	—
QNaN	0x3E	—	—	—
—	0x15	1.625	1	3.25
—	0x22	0.25	0	-0.25
One	0x08	1.0	0	1.0
Smallest denormalized > 0	0x01	0.125	0	0.125
Largest normalized > 0				

Problem 2.8

Largest normalized > 0

Problem 2.8

Largest normalized > 0

0

Problem 2.8

Largest normalized > 0

010

Problem 2.8

Largest normalized > 0

010111

Problem 2.8

Largest normalized > 0

010111

Bias is 1 E is 1

Problem 2.8

Largest normalized > 0

010111

Bias is 1 E is 1

$$1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} = 1.875$$

Problem 2.8

Largest normalized > 0

010111

Bias is 1 E is 1

$$1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} = 1.875$$

$$1 * 2^1 * 1.875 = 3.75$$

Problem 2

Description	Hex	M	E	Value
Negative zero	0x20	—	—	—
Positive infinity	0x18	—	—	—
QNaN	0x3E	—	—	—
—	0x15	1.625	1	3.25
—	0x22	0.25	0	-0.25
One	0x08	1.0	0	1.0
Smallest denormalized > 0	0x01	0.125	0	0.125
Largest normalized > 0	0x17	1.875	1	3.75

Problem 3

```
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jge .L6
  movl %edx, %eax
.L6:
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %eax, %edx
  jb .L9
  movl %edx, %eax
.L9:
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine ____.

Problem 3

```
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jge .L6
  movl %edx, %eax
.L6:
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %eax, %edx
  jb .L9
  movl %edx, %eax
.L9:
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine ____.

Problem 3

```
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax
  jge .L6
  movl %edx, %eax
.L6:
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %eax, %edx
  jb .L9
  movl %edx, %eax
.L9:
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine ____.

Problem 3

```
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6
  movl %edx, %eax
.L6:
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %eax, %edx
  jb .L9
  movl %edx, %eax
.L9:
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine ____.

Problem 3

```
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax
.L6:
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %eax, %edx
  jb .L9
  movl %edx, %eax
.L9:
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine ____.

Problem 3

```
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %eax, %edx
  jb .L9
  movl %edx, %eax
.L9:
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine ____.

Problem 3

```
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:; return b if b is the largest
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %eax, %edx
  jb .L9
  movl %edx, %eax
.L9:
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine ____.

Problem 3

```c
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```asm
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:; return b if b is the largest
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax
  cmpl %eax, %edx
  jb .L9
  movl %edx, %eax
.L9:
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine ____.

Problem 3

```
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:; return b if b is the largest
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %eax, %edx
  jb .L9
  movl %edx, %eax
.L9:
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine ____.

Problem 3

```c
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:; return b if b is the largest
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %eax, %edx; a - b
  jb .L9
  movl %edx, %eax
.L9:
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine ____.

Problem 3

```c
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```asm
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:; return b if b is the largest
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %eax, %edx; a - b
  jb .L9; jmp if a < b
  movl %edx, %eax
.L9:
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine ____.

Problem 3

```
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:; return b if b is the largest
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %eax, %edx; a - b
  jb .L9; jmp if a < b
  movl %edx, %eax; return a if a is the largest
.L9:
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine ____.

Problem 3

```
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:; return b if b is the largest
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %eax, %edx; a - b
  jb .L9; jmp if a < b
  movl %edx, %eax; return a if a is the largest
.L9:; return b if b is the largest
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine ____.

Problem 3

```
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:; return b if b is the largest
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %eax, %edx; a - b
  jb .L9; jmp if a < b
  movl %edx, %eax; return a if a is the largest
.L9:; return b if b is the largest
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine asm2.

Problem 3

```c
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:; return b if b is the largest
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %eax, %edx; a - b
  jb .L9; jmp if a < b
  movl %edx, %eax; return a if a is the largest
.L9:; return b if b is the largest
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine asm2.

Problem 3

```c
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:; return b if b is the largest
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %eax, %edx; a - b
  jb .L9; jmp if a < b
  movl %edx, %eax; return a if a is the largest
.L9:; return b if b is the largest
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine asm2.

Problem 3

```c
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:; return b if b is the largest
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %eax, %edx; a - b
  jb .L9; jmp if a < b
  movl %edx, %eax; return a if a is the largest
.L9:; return b if b is the largest
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jle .L2
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine asm2.

Problem 3

```c
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:; return b if b is the largest
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %eax, %edx; a - b
  jb .L9; jmp if a < b
  movl %edx, %eax; return a if a is the largest
.L9:; return b if b is the largest
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jle .L2; jmp if b <= a
  movl %edx, %eax
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine asm2.

Problem 3

```c
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:; return b if b is the largest
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %eax, %edx; a - b
  jb .L9; jmp if a < b
  movl %edx, %eax; return a if a is the largest
.L9:; return b if b is the largest
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jle .L2; jmp if b <= a
  movl %edx, %eax; return a if a is the smallest
.L2:
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine asm2.

Problem 3

```c
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:; return b if b is the largest
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %eax, %edx; a - b
  jb .L9; jmp if a < b
  movl %edx, %eax; return a if a is the largest
.L9:; return b if b is the largest
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jle .L2; jmp if b <= a
  movl %edx, %eax; return a if a is the smallest
.L2:; return b if b is the smallest
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine ____.
C function baz2 corresponds to assembly-code routine ____.
C function baz3 corresponds to assembly-code routine asm2.

Problem 3

```c
int baz1(int a, int b) {
  if(b < a)
    return b;
  return a;
}

int baz2(int a, int b) {
  if(a < b)
    return a;
  return b;
}

int baz3(int a, int b) {
  unsigned ua = (unsigned) a;
  if (ua < b)
    return b;
  return ua;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jge .L6; jmp if b >= a
  movl %edx, %eax; return a if a is the largest
.L6:; return b if b is the largest
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %eax, %edx; a - b
  jb .L9; jmp if a < b
  movl %edx, %eax; return a if a is the largest
.L9:; return b if b is the largest
  popl %ebp
  ret

asm3:
  pushl %ebp
  movl %esp, %ebp
  movl 8(%ebp), %edx; a -> %edx
  movl 12(%ebp), %eax; b -> %eax
  cmpl %edx, %eax; b - a
  jle .L2; jmp if b <= a
  movl %edx, %eax; return a if a is the smallest
.L2:; return b if b is the smallest
  popl %ebp
  ret
```

C function baz1 corresponds to assembly-code routine asm3.
C function baz2 corresponds to assembly-code routine asm3.
C function baz3 corresponds to assembly-code routine asm2.

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx
  pushl %esi
  movl 8(%ebp), %esi
  testl %ecx, %ecx
  jle .L4
  xorl %edx, %edx
.L5:
  incl %edx
  imull %esi, %eax
  cmpl %edx, %ecx
  jne .L5
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = _, result = _; _____; i++)
   ___________;
 return result;
}
```

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx
  pushl %esi
  movl 8(%ebp), %esi
  testl %ecx, %ecx
  jle .L4
  xorl %edx, %edx
.L5:
  incl %edx
  imull %esi, %eax
  cmpl %edx, %ecx
  jne .L5
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = _, result = _; _____; i++)
   ___________;
 return result;
}
```

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx
  pushl %esi
  movl 8(%ebp), %esi
  testl %ecx, %ecx
  jle .L4
  xorl %edx, %edx
.L5:
  incl %edx
  imull %esi, %eax
  cmpl %edx, %ecx
  jne .L5
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = _, result = 1; _____; i++)
   ___________;
 return result;
}
```

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx; y -> %ecx
  pushl %esi
  movl 8(%ebp), %esi
  testl %ecx, %ecx
  jle .L4
  xorl %edx, %edx
.L5:
  incl %edx
  imull %esi, %eax
  cmpl %edx, %ecx
  jne .L5
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = _, result = 1; _____; i++)
   ___________;
 return result;
}
```

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx; y -> %ecx
  pushl %esi;save  %esi
  movl 8(%ebp), %esi
  testl %ecx, %ecx
  jle .L4
  xorl %edx, %edx
.L5:
  incl %edx
  imull %esi, %eax
  cmpl %edx, %ecx
  jne .L5
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = _, result = 1; _____; i++)
   ___________;
 return result;
}
```

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx; y -> %ecx
  pushl %esi;save  %esi
  movl 8(%ebp), %esi; x -> %esi
  testl %ecx, %ecx
  jle .L4
  xorl %edx, %edx
.L5:
  incl %edx
  imull %esi, %eax
  cmpl %edx, %ecx
  jne .L5
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = _, result = 1; _____; i++)
   ___________;
 return result;
}
```

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx; y -> %ecx
  pushl %esi;save  %esi
  movl 8(%ebp), %esi; x -> %esi
  testl %ecx, %ecx; y && y
  jle .L4; jmp y <= 0
  xorl %edx, %edx
.L5:
  incl %edx
  imull %esi, %eax
  cmpl %edx, %ecx
  jne .L5
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = _, result = 1; _____; i++)
   ___________;
 return result;
}
```

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx; y -> %ecx
  pushl %esi;save  %esi
  movl 8(%ebp), %esi; x -> %esi
  testl %ecx, %ecx; y && y
  jle .L4; jmp y <= 0
  xorl %edx, %edx
.L5:
  incl %edx
  imull %esi, %eax
  cmpl %edx, %ecx
  jne .L5
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = _, result = 1; _____; i++)
   ___________;
 return result;
}
```

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx; y -> %ecx
  pushl %esi;save  %esi
  movl 8(%ebp), %esi; x -> %esi
  testl %ecx, %ecx; y && y
  jle .L4; jmp y <= 0
  xorl %edx, %edx; i = 0 -> %edx
.L5:
  incl %edx
  imull %esi, %eax
  cmpl %edx, %ecx
  jne .L5
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = _, result = 1; _____; i++)
   ___________;
 return result;
}
```

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx; y -> %ecx
  pushl %esi;save  %esi
  movl 8(%ebp), %esi; x -> %esi
  testl %ecx, %ecx; y && y
  jle .L4; jmp y <= 0
  xorl %edx, %edx; i = 0 -> %edx
.L5:
  incl %edx
  imull %esi, %eax
  cmpl %edx, %ecx
  jne .L5
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = 0, result = 1; _____; i++)
   ____________;
 return result;
}
```

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx; y -> %ecx
  pushl %esi;save  %esi
  movl 8(%ebp), %esi; x -> %esi
  testl %ecx, %ecx; y && y
  jle .L4; jmp y <= 0
  xorl %edx, %edx; i = 0 -> %edx
.L5:
  incl %edx; i++
  imull %esi, %eax
  cmpl %edx, %ecx
  jne .L5
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = 0, result = 1; _____; i++)
   ____________;
 return result;
}
```

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx; y -> %ecx
  pushl %esi;save  %esi
  movl 8(%ebp), %esi; x -> %esi
  testl %ecx, %ecx; y && y
  jle .L4; jmp y <= 0
  xorl %edx, %edx; i = 0 -> %edx
.L5:
  incl %edx; i++
  imull %esi, %eax; result *= x
  cmpl %edx, %ecx
  jne .L5
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = 0, result = 1; _____; i++)
   ____________;
 return result;
}
```

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx; y -> %ecx
  pushl %esi;save  %esi
  movl 8(%ebp), %esi; x -> %esi
  testl %ecx, %ecx; y && y
  jle .L4; jmp y <= 0
  xorl %edx, %edx; i = 0 -> %edx
.L5:
  incl %edx; i++
  imull %esi, %eax; result *= x
  cmpl %edx, %ecx
  jne .L5
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = 0, result = 1; _____; i++)
   result *= x;
 return result;
}
```

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx; y -> %ecx
  pushl %esi;save  %esi
  movl 8(%ebp), %esi; x -> %esi
  testl %ecx, %ecx; y && y
  jle .L4; jmp y <= 0
  xorl %edx, %edx; i = 0 -> %edx
.L5:
  incl %edx; i++
  imull %esi, %eax; result *= x
  cmpl %edx, %ecx; y - i
  jne .L5; jmp y != i
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = 0, result = 1; _____; i++)
   result *= x;
 return result;
}
```

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx; y -> %ecx
  pushl %esi;save  %esi
  movl 8(%ebp), %esi; x -> %esi
  testl %ecx, %ecx; y && y
  jle .L4; jmp y <= 0
  xorl %edx, %edx; i = 0 -> %edx
.L5:
  incl %edx; i++
  imull %esi, %eax; result *= x
  cmpl %edx, %ecx; y - i
  jne .L5; jmp y != i    i < y
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = 0, result = 1; _____; i++)
   result *= x;
 return result;
}
```

Problem 4

```
bar:
  pushl %ebp
  movl $1, %eax
  movl %esp, %ebp
  movl 12(%ebp), %ecx; y -> %ecx
  pushl %esi;save  %esi
  movl 8(%ebp), %esi; x -> %esi
  testl %ecx, %ecx; y && y
  jle .L4; jmp y <= 0
  xorl %edx, %edx; i = 0 -> %edx
.L5:
  incl %edx; i++
  imull %esi, %eax; result *= x
  cmpl %edx, %ecx; y - i
  jne .L5; jmp y != i    i < y
.L4:
  popl %esi
  leave
  ret
```

```
int bar(int x, int y) {
  int i, result;
  for (i = 0, result = 1; i < y; i++)
   result *= x;
 return result;
}
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}

scale:
  pushl %ebp
  movl %esp, %ebp
  subl $8, %esp
  movl 8(%ebp), %ecx
  movl %ebx, (%esp)
  movl 12(%ebp), %eax
  movl 16(%ebp), %edx
  movl %esi, 4(%esp)
  leal (%eax,%ecx,14), %ebx
  movl arr1(,%ebx,4), %esi
  leal (%eax,%eax,2), %eax
  leal (%ecx,%eax,4), %eax
  imull %edx, %esi
  imull arr2(,%eax,4), %edx
  movl %esi, arr1(,%ebx,4)
  movl (%esp), %ebx
  movl %edx, arr2(,%eax,4)
  movl 4(%esp), %esi
  movl %ebp, %esp
  popl %ebp
  ret

M =
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}

scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp
  movl 8(%ebp), %ecx
  movl %ebx, (%esp)
  movl 12(%ebp), %eax
  movl 16(%ebp), %edx
  movl %esi, 4(%esp)
  leal (%eax,%ecx,14), %ebx
  movl arr1(,%ebx,4), %esi
  leal (%eax,%eax,2), %eax
  leal (%ecx,%eax,4), %eax
  imull %edx, %esi
  imull arr2(,%eax,4), %edx
  movl %esi, arr1(,%ebx,4)
  movl (%esp), %ebx
  movl %edx, arr2(,%eax,4)
  movl 4(%esp), %esi
  movl %ebp, %esp
  popl %ebp
  ret

M =
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}

scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax
  movl 16(%ebp), %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx
  movl arr1(,%ebx,4), %esi
  leal (%eax,%eax,2), %eax
  leal (%ecx,%eax,4), %eax
  imull %edx, %esi
  imull arr2(,%eax,4), %edx
  movl %esi, arr1(,%ebx,4)
  movl (%esp), %ebx
  movl %edx, arr2(,%eax,4)
  movl 4(%esp), %esi
  movl %ebp, %esp
  popl %ebp
  ret

M =
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}

scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax
  movl 16(%ebp), %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx
  movl arr1(,%ebx,4), %esi
  leal (%eax,%eax,2), %eax
  leal (%ecx,%eax,4), %eax
  imull %edx, %esi
  imull arr2(,%eax,4), %edx
  movl %esi, arr1(,%ebx,4)
  movl (%esp), %ebx
  movl %edx, arr2(,%eax,4)
  movl 4(%esp), %esi
  movl %ebp, %esp
  popl %ebp
  ret

M =
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}

scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax; j -> %eax
  movl 16(%ebp), %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx
  movl arr1(,%ebx,4), %esi
  leal (%eax,%eax,2), %eax
  leal (%ecx,%eax,4), %eax
  imull %edx, %esi
  imull arr2(,%eax,4), %edx
  movl %esi, arr1(,%ebx,4)
  movl (%esp), %ebx
  movl %edx, arr2(,%eax,4)
  movl 4(%esp), %esi
  movl %ebp, %esp
  popl %ebp
  ret

M =
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}

scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax; j -> %eax
  movl 16(%ebp), %edx; s -> %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx
  movl arr1(,%ebx,4), %esi
  leal (%eax,%eax,2), %eax
  leal (%ecx,%eax,4), %eax
  imull %edx, %esi
  imull arr2(,%eax,4), %edx
  movl %esi, arr1(,%ebx,4)
  movl (%esp), %ebx
  movl %edx, arr2(,%eax,4)
  movl 4(%esp), %esi
  movl %ebp, %esp
  popl %ebp
  ret

M =
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}

scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax; j -> %eax
  movl 16(%ebp), %edx; s -> %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx; %ebx = j + (i*14)
  movl arr1(,%ebx,4), %esi
  leal (%eax,%eax,2), %eax
  leal (%ecx,%eax,4), %eax
  imull %edx, %esi
  imull arr2(,%eax,4), %edx
  movl %esi, arr1(,%ebx,4)
  movl (%esp), %ebx
  movl %edx, arr2(,%eax,4)
  movl 4(%esp), %esi
  movl %ebp, %esp
  popl %ebp
  ret

M =
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}

scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax; j -> %eax
  movl 16(%ebp), %edx; s -> %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx; %ebx = j + (i*14)
  movl arr1(,%ebx,4), %esi; %esi = arr1[i][j]
  leal (%eax,%eax,2), %eax
  leal (%ecx,%eax,4), %eax
  imull %edx, %esi
  imull arr2(,%eax,4), %edx
  movl %esi, arr1(,%ebx,4)
  movl (%esp), %ebx
  movl %edx, arr2(,%eax,4)
  movl 4(%esp), %esi
  movl %ebp, %esp
  popl %ebp
  ret

M =
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}

scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax; j -> %eax
  movl 16(%ebp), %edx; s -> %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx; %ebx = j + (i*14)
  movl arr1(,%ebx,4), %esi; %esi = arr1[i][j]
  leal (%eax,%eax,2), %eax; %eax = j + (2*j), %eax = 3j
  leal (%ecx,%eax,4), %eax
  imull %edx, %esi
  imull arr2(,%eax,4), %edx
  movl %esi, arr1(,%ebx,4)
  movl (%esp), %ebx
  movl %edx, arr2(,%eax,4)
  movl 4(%esp), %esi
  movl %ebp, %esp
  popl %ebp
  ret

M =
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}

scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax; j -> %eax
  movl 16(%ebp), %edx; s -> %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx; %ebx = j + (i*14)
  movl arr1(,%ebx,4), %esi; %esi = arr1[i][j]
  leal (%eax,%eax,2), %eax; %eax = j + (2*j), %eax = 3j
  leal (%ecx,%eax,4), %eax; %eax = i + (4 * 3 * j), %eax = i + 12j
  imull %edx, %esi
  imull arr2(,%eax,4), %edx
  movl %esi, arr1(,%ebx,4)
  movl (%esp), %ebx
  movl %edx, arr2(,%eax,4)
  movl 4(%esp), %esi
  movl %ebp, %esp
  popl %ebp
  ret

M =
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}

scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax; j -> %eax
  movl 16(%ebp), %edx; s -> %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx; %ebx = j + (i*14)
  movl arr1(,%ebx,4), %esi; %esi = arr1[i][j]
  leal (%eax,%eax,2), %eax; %eax = j + (2*j), %eax = 3j
  leal (%ecx,%eax,4), %eax; %eax = i + (4 * 3 * j), %eax = i + 12j
  imull %edx, %esi; %esi = %esi * s, %esi = (arr1[i][j])
  imull arr2(,%eax,4), %edx
  movl %esi, arr1(,%ebx,4)
  movl (%esp), %ebx
  movl %edx, arr2(,%eax,4)
  movl 4(%esp), %esi
  movl %ebp, %esp
  popl %ebp
  ret

M =
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}
```

```
scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax; j -> %eax
  movl 16(%ebp), %edx; s -> %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx; %ebx = j + (i*14)
  movl arr1(,%ebx,4), %esi; %esi = arr1[i][j]
  leal (%eax,%eax,2), %eax; %eax = j + (2*j), %eax = 3j
  leal (%ecx,%eax,4), %eax; %eax = i + (4 * 3 * j), %eax = i + 12j
  imull %edx, %esi; %esi = %esi * s, %esi = (arr1[i][j])
  imull arr2(,%eax,4), %edx; %edx = arr2[j][i] * s
  movl %esi, arr1(,%ebx,4)
  movl (%esp), %ebx
  movl %edx, arr2(,%eax,4)
  movl 4(%esp), %esi
  movl %ebp, %esp
  popl %ebp
  ret
```

```
M =
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}
```

```
scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax; j -> %eax
  movl 16(%ebp), %edx; s -> %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx; %ebx = j + (i*14)
  movl arr1(,%ebx,4), %esi; %esi = arr1[i][j]
  leal (%eax,%eax,2), %eax; %eax = j + (2*j), %eax = 3j
  leal (%ecx,%eax,4), %eax; %eax = i + (4 * 3 * j), %eax = i + 12j
  imull %edx, %esi; %esi = %esi * s, %esi = (arr1[i][j])
  imull arr2(,%eax,4), %edx; %edx = arr2[j][i] * s
  movl %esi, arr1(,%ebx,4); arr1[i][j] = %esi
  movl (%esp), %ebx
  movl %edx, arr2(,%eax,4)
  movl 4(%esp), %esi
  movl %ebp, %esp
  popl %ebp
  ret
```

M =
N =

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}
```

```
scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax; j -> %eax
  movl 16(%ebp), %edx; s -> %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx; %ebx = j + (i*14)
  movl arr1(,%ebx,4), %esi; %esi = arr1[i][j]
  leal (%eax,%eax,2), %eax; %eax = j + (2*j), %eax = 3j
  leal (%ecx,%eax,4), %eax; %eax = i + (4 * 3 * j), %eax = i + 12j
  imull %edx, %esi; %esi = %esi * s, %esi = (arr1[i][j])
  imull arr2(,%eax,4), %edx; %edx = arr2[j][i] * s
  movl %esi, arr1(,%ebx,4); arr1[i][j] = %esi
  movl (%esp), %ebx; restore %ebx
  movl %edx, arr2(,%eax,4)
  movl 4(%esp), %esi
  movl %ebp, %esp
  popl %ebp
  ret
```

```
M =
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}

scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax; j -> %eax
  movl 16(%ebp), %edx; s -> %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx; %ebx = j + (i*14)
  movl arr1(,%ebx,4), %esi; %esi = arr1[i][j]
  leal (%eax,%eax,2), %eax; %eax = j + (2*j), %eax = 3j
  leal (%ecx,%eax,4), %eax; %eax = i + (4 * 3 * j), %eax = i + 12j
  imull %edx, %esi; %esi = %esi * s, %esi = (arr1[i][j])
  imull arr2(,%eax,4), %edx; %edx = arr2[j][i] * s
  movl %esi, arr1(,%ebx,4); arr1[i][j] = %esi
  movl (%esp), %ebx; restore %ebx
  movl %edx, arr2(,%eax,4); arr2[j][i] = %edx
  movl 4(%esp), %esi
  movl %ebp, %esp
  popl %ebp
  ret

M =
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}
```

```
scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax; j -> %eax
  movl 16(%ebp), %edx; s -> %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx; %ebx = j + (i*14)
  movl arr1(,%ebx,4), %esi; %esi = arr1[i][j]
  leal (%eax,%eax,2), %eax; %eax = j + (2*j), %eax = 3j
  leal (%ecx,%eax,4), %eax; %eax = i + (4 * 3 * j), %eax = i + 12j
  imull %edx, %esi; %esi = %esi * s, %esi = (arr1[i][j])
  imull arr2(,%eax,4), %edx; %edx = arr2[j][i] * s
  movl %esi, arr1(,%ebx,4); arr1[i][j] = %esi
  movl (%esp), %ebx; restore %ebx
  movl %edx, arr2(,%eax,4); arr2[j][i] = %edx
  movl 4(%esp), %esi; restore %esi
  movl %ebp, %esp
  popl %ebp
  ret
```

M =
N =

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}

scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax; j -> %eax
  movl 16(%ebp), %edx; s -> %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx; %ebx = j + (i*14)
  movl arr1(,%ebx,4), %esi; %esi = arr1[i][j]
  leal (%eax,%eax,2), %eax; %eax = j + (2*j), %eax = 3j
  leal (%ecx,%eax,4), %eax; %eax = i + (4 * 3 * j), %eax = i + 12j
  imull %edx, %esi; %esi = %esi * s, %esi = (arr1[i][j])
  imull arr2(,%eax,4), %edx; %edx = arr2[j][i] * s
  movl %esi, arr1(,%ebx,4); arr1[i][j] = %esi
  movl (%esp), %ebx; restore %ebx
  movl %edx, arr2(,%eax,4); arr2[j][i] = %edx
  movl 4(%esp), %esi; restore %esi
  movl %ebp, %esp; postamble
  popl %ebp; postamble
  ret

M =
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}

scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax; j -> %eax
  movl 16(%ebp), %edx; s -> %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx; %ebx = j + (i*14)
  movl arr1(,%ebx,4), %esi; %esi = arr1[i][j]
  leal (%eax,%eax,2), %eax; %eax = j + (2*j), %eax = 3j
  leal (%ecx,%eax,4), %eax; %eax = i + (4 * 3 * j), %eax = i + 12j
  imull %edx, %esi; %esi = %esi * s, %esi = (arr1[i][j])
  imull arr2(,%eax,4), %edx; %edx = arr2[j][i] * s
  movl %esi, arr1(,%ebx,4); arr1[i][j] = %esi
  movl (%esp), %ebx; restore %ebx
  movl %edx, arr2(,%eax,4); arr2[j][i] = %edx
  movl 4(%esp), %esi; restore %esi
  movl %ebp, %esp; postamble
  popl %ebp; postamble
  ret

M = 12
N =
```

Problem 5

```
int arr1[M][N];
int arr2[N][M];

void scale(int i, int j, int s) {
  arr1[i][j] *= s;
  arr2[j][i] *= s;
}

scale:
  pushl %ebp; preamble
  movl %esp, %ebp; preamble
  subl $8, %esp; callee save
  movl 8(%ebp), %ecx; i -> %ecx
  movl %ebx, (%esp); callee save %ebx
  movl 12(%ebp), %eax; j -> %eax
  movl 16(%ebp), %edx; s -> %edx
  movl %esi, 4(%esp); callee save %esi
  leal (%eax,%ecx,14), %ebx; %ebx = j + (i*14)
  movl arr1(,%ebx,4), %esi; %esi = arr1[i][j]
  leal (%eax,%eax,2), %eax; %eax = j + (2*j), %eax = 3j
  leal (%ecx,%eax,4), %eax; %eax = i + (4 * 3 * j), %eax = i + 12j
  imull %edx, %esi; %esi = %esi * s, %esi = (arr1[i][j])
  imull arr2(,%eax,4), %edx; %edx = arr2[j][i] * s
  movl %esi, arr1(,%ebx,4); arr1[i][j] = %esi
  movl (%esp), %ebx; restore %ebx
  movl %edx, arr2(,%eax,4); arr2[j][i] = %edx
  movl 4(%esp), %esi; restore %esi
  movl %ebp, %esp; postamble
  popl %ebp; postamble
  ret

M = 12
N = 14
```

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx
  movl 8(%ebp), %edx
  movl 16(%ebp), %eax
  movl (%ecx), %ecx
  movl (%eax), %eax
  addl %ecx, (%edx)
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax
  movl 8(%ebp), %edx
  movl (%eax), %eax
  addl (%edx), %eax
  movl 16(%ebp), %edx
  popl %ebp
  addl (%edx), %eax
  ret
```

Assembly-code routine asm1 corresponds to C function _____

Assembly-code routine asm2 corresponds to C function _____

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx; %ecx = ptr2
  movl 8(%ebp), %edx
  movl 16(%ebp), %eax
  movl (%ecx), %ecx
  movl (%eax), %eax
  addl %ecx, (%edx)
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax
  movl 8(%ebp), %edx
  movl (%eax), %eax
  addl (%edx), %eax
  movl 16(%ebp), %edx
  popl %ebp
  addl (%edx), %eax
  ret
```

Assembly-code routine asm1 corresponds to C function _____

Assembly-code routine asm2 corresponds to C function _____

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx; %ecx = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl 16(%ebp), %eax
  movl (%ecx), %ecx
  movl (%eax), %eax
  addl %ecx, (%edx)
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax
  movl 8(%ebp), %edx
  movl (%eax), %eax
  addl (%edx), %eax
  movl 16(%ebp), %edx
  popl %ebp
  addl (%edx), %eax
  ret
```

Assembly-code routine asm1 corresponds to C function _____

Assembly-code routine asm2 corresponds to C function _____

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx; %ecx = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl 16(%ebp), %eax; %eax = ptr3
  movl (%ecx), %ecx
  movl (%eax), %eax
  addl %ecx, (%edx)
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax
  movl 8(%ebp), %edx
  movl (%eax), %eax
  addl (%edx), %eax
  movl 16(%ebp), %edx
  popl %ebp
  addl (%edx), %eax
  ret
```

Assembly-code routine asm1 corresponds to C function _____

Assembly-code routine asm2 corresponds to C function _____

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx; %ecx = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl 16(%ebp), %eax; %eax = ptr3
  movl (%ecx), %ecx; %ecx = *ptr2
  movl (%eax), %eax; %eax = *ptr3
  addl %ecx, (%edx)
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax
  movl 8(%ebp), %edx
  movl (%eax), %eax
  addl (%edx), %eax
  movl 16(%ebp), %edx
  popl %ebp
  addl (%edx), %eax
  ret
```

Assembly-code routine asm1 corresponds to C function _____

Assembly-code routine asm2 corresponds to C function _____

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx; %ecx = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl 16(%ebp), %eax; %eax = ptr3
  movl (%ecx), %ecx; %ecx = *ptr2
  movl (%eax), %eax; %eax = *ptr3
  addl %ecx, (%edx); *ptr1 += *ptr2
  popl %ebp
  ret

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax
  movl 8(%ebp), %edx
  movl (%eax), %eax
  addl (%edx), %eax
  movl 16(%ebp), %edx
  popl %ebp
  addl (%edx), %eax
  ret
```

Assembly-code routine asm1 corresponds to C function _____

Assembly-code routine asm2 corresponds to C function _____

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx; %ecx = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl 16(%ebp), %eax; %eax = ptr3
  movl (%ecx), %ecx; %ecx = *ptr2
  movl (%eax), %eax; %eax = *ptr3
  addl %ecx, (%edx); *ptr1 += *ptr2
  popl %ebp
  ret; return *ptr3

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax
  movl 8(%ebp), %edx
  movl (%eax), %eax
  addl (%edx), %eax
  movl 16(%ebp), %edx
  popl %ebp
  addl (%edx), %eax
  ret
```

Assembly-code routine asm1 corresponds to C function _____

Assembly-code routine asm2 corresponds to C function _____

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx; %ecx = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl 16(%ebp), %eax; %eax = ptr3
  movl (%ecx), %ecx; %ecx = *ptr2
  movl (%eax), %eax; %eax = *ptr3
  addl %ecx, (%edx); *ptr1 += *ptr2
  popl %ebp
  ret; return *ptr3

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax
  movl 8(%ebp), %edx
  movl (%eax), %eax
  addl (%edx), %eax
  movl 16(%ebp), %edx
  popl %ebp
  addl (%edx), %eax
  ret
```

Assembly-code routine asm1 corresponds to C function foo5

Assembly-code routine asm2 corresponds to C function _____

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx; %ecx = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl 16(%ebp), %eax; %eax = ptr3
  movl (%ecx), %ecx; %ecx = *ptr2
  movl (%eax), %eax; %eax = *ptr3
  addl %ecx, (%edx); *ptr1 += *ptr2
  popl %ebp
  ret; return *ptr3

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax; %eax = ptr2
  movl 8(%ebp), %edx
  movl (%eax), %eax
  addl (%edx), %eax
  movl 16(%ebp), %edx
  popl %ebp
  addl (%edx), %eax
  ret
```

Assembly-code routine asm1 corresponds to C function foo5

Assembly-code routine asm2 corresponds to C function _____

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx; %ecx = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl 16(%ebp), %eax; %eax = ptr3
  movl (%ecx), %ecx; %ecx = *ptr2
  movl (%eax), %eax; %eax = *ptr3
  addl %ecx, (%edx); *ptr1 += *ptr2
  popl %ebp
  ret; return *ptr3

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax; %eax = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl (%eax), %eax
  addl (%edx), %eax
  movl 16(%ebp), %edx
  popl %ebp
  addl (%edx), %eax
  ret
```

Assembly-code routine asm1 corresponds to C function foo5

Assembly-code routine asm2 corresponds to C function _____

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx; %ecx = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl 16(%ebp), %eax; %eax = ptr3
  movl (%ecx), %ecx; %ecx = *ptr2
  movl (%eax), %eax; %eax = *ptr3
  addl %ecx, (%edx); *ptr1 += *ptr2
  popl %ebp
  ret; return *ptr3

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax; %eax = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl (%eax), %eax; %eax = *ptr2
  addl (%edx), %eax
  movl 16(%ebp), %edx
  popl %ebp
  addl (%edx), %eax
  ret
```

Assembly-code routine asm1 corresponds to C function foo5

Assembly-code routine asm2 corresponds to C function _____

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx; %ecx = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl 16(%ebp), %eax; %eax = ptr3
  movl (%ecx), %ecx; %ecx = *ptr2
  movl (%eax), %eax; %eax = *ptr3
  addl %ecx, (%edx); *ptr1 += *ptr2
  popl %ebp
  ret; return *ptr3

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax; %eax = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl (%eax), %eax; %eax = *ptr2
  addl (%edx), %eax; tmp = *ptr1 + *ptr2, %eax becomes tmp
  movl 16(%ebp), %edx
  popl %ebp
  addl (%edx), %eax
  ret
```

Assembly-code routine asm1 corresponds to C function foo5

Assembly-code routine asm2 corresponds to C function _____

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx; %ecx = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl 16(%ebp), %eax; %eax = ptr3
  movl (%ecx), %ecx; %ecx = *ptr2
  movl (%eax), %eax; %eax = *ptr3
  addl %ecx, (%edx); *ptr1 += *ptr2
  popl %ebp
  ret; return *ptr3

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax; %eax = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl (%eax), %eax; %eax = *ptr2
  addl (%edx), %eax; tmp = *ptr1 + *ptr2, %eax becomes tmp
  movl 16(%ebp), %edx; %edx = ptr3
  popl %ebp
  addl (%edx), %eax
  ret
```

Assembly-code routine asm1 corresponds to C function foo5

Assembly-code routine asm2 corresponds to C function _____

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx; %ecx = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl 16(%ebp), %eax; %eax = ptr3
  movl (%ecx), %ecx; %ecx = *ptr2
  movl (%eax), %eax; %eax = *ptr3
  addl %ecx, (%edx); *ptr1 += *ptr2
  popl %ebp
  ret; return *ptr3

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax; %eax = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl (%eax), %eax; %eax = *ptr2
  addl (%edx), %eax; tmp = *ptr1 + *ptr2, %eax becomes tmp
  movl 16(%ebp), %edx; %edx = ptr3
  popl %ebp
  addl (%edx), %eax; tmp += *ptr3, %eax is tmp
  ret
```

Assembly-code routine asm1 corresponds to C function foo5

Assembly-code routine asm2 corresponds to C function _____

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx; %ecx = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl 16(%ebp), %eax; %eax = ptr3
  movl (%ecx), %ecx; %ecx = *ptr2
  movl (%eax), %eax; %eax = *ptr3
  addl %ecx, (%edx); *ptr1 += *ptr2
  popl %ebp
  ret; return *ptr3

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax; %eax = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl (%eax), %eax; %eax = *ptr2
  addl (%edx), %eax; tmp = *ptr1 + *ptr2, %eax becomes tmp
  movl 16(%ebp), %edx; %edx = ptr3
  popl %ebp
  addl (%edx), %eax; tmp += *ptr3, %eax is tmp
  ret; return %eax, *ptr1 + *ptr2 + *ptr3
```

Assembly-code routine asm1 corresponds to C function foo5

Assembly-code routine asm2 corresponds to C function _____

Problem 6

```
int foo1(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  int y = *ptr2;
  int z = *ptr3;
  return x + y + z;
}

int foo2(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr2;
  int y = *ptr3;
  int z = *ptr1;
  return x + y + z;
}

int foo3(int *ptr1, int *ptr2, int *ptr3) {
  int y = *ptr2;
  *ptr1 += *ptr3;
  return y;
}

int foo4(int *ptr1, int *ptr2, int *ptr3) {
  int x = *ptr1;
  *ptr3 += *ptr2;
  return x;
}

int foo5(int *ptr1, int *ptr2, int *ptr3) {
  int z = *ptr3;
  *ptr1 += *ptr2;
  return z;
}
```

```
asm1:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %ecx; %ecx = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl 16(%ebp), %eax; %eax = ptr3
  movl (%ecx), %ecx; %ecx = *ptr2
  movl (%eax), %eax; %eax = *ptr3
  addl %ecx, (%edx); *ptr1 += *ptr2
  popl %ebp
  ret; return *ptr3

asm2:
  pushl %ebp
  movl %esp, %ebp
  movl 12(%ebp), %eax; %eax = ptr2
  movl 8(%ebp), %edx; %edx = ptr1
  movl (%eax), %eax; %eax = *ptr2
  addl (%edx), %eax; tmp = *ptr1 + *ptr2, %eax becomes tmp
  movl 16(%ebp), %edx; %edx = ptr3
  popl %ebp
  addl (%edx), %eax; tmp += *ptr3, %eax is tmp
  ret; return %eax, *ptr1 + *ptr2 + *ptr3
```

Assembly-code routine asm1 corresponds to C function foo5
Assembly-code routine asm2 corresponds to C function foo1 or foo2

Problem 7

```
struct s1 {
  char a[3];
  union u1 b;
  int c;
};

struct s2 {
  struct s1 *d;
  char e;
  int f[4];
  struct s2 *g;
};

union u1 {
  struct s1 *h;
  struct s2 *i;
  char j;
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 8(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 12(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->____________________;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b;
  int c;
};

struct s2 {
  struct s1 *d;
  char e;
  int f[4];
  struct s2 *g;
};

union u1 {
  struct s1 *h;
  struct s2 *i;
  char j;
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 8(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 12(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->____________________;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c;
};

struct s2 {
  struct s1 *d;
  char e;
  int f[4];
  struct s2 *g;
};

union u1 {
  struct s1 *h;
  struct s2 *i;
  char j;
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 8(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 12(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->____________________;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d;
  char e;
  int f[4];
  struct s2 *g;
};

union u1 {
  struct s1 *h;
  struct s2 *i;
  char j;
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 8(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 12(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->____________________;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e;
  int f[4];
  struct s2 *g;
};

union u1 {
  struct s1 *h;
  struct s2 *i;
  char j;
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 8(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 12(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->____________________;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4];
  struct s2 *g;
};

union u1 {
  struct s1 *h;
  struct s2 *i;
  char j;
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 8(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 12(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->____________________;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g;
};

union u1 {
  struct s1 *h;
  struct s2 *i;
  char j;
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 8(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 12(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->____________________;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h;
  struct s2 *i;
  char j;
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 8(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 12(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->____________________;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i;
  char j;
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 8(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 12(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->____________________;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j;
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 8(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 12(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->____________________;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 8(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 12(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->____________________;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 12(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->____________________;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax; %eax = x->c
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 12(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->____________________;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax; %eax = x->c
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 12(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->c;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax; %eax = x->c
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 12(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->c;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax; %eax = x->c
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 12(%eax),%eax; %eax = x->f[1]
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->c;
}
```

```
int proc2(struct s2 *x) {
return x->____________________;
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax; %eax = x->c
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 12(%eax),%eax; %eax = x->f[1]
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->c;
}
```

```
int proc2(struct s2 *x) {
return x->f[1];
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax; %eax = x->c
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 12(%eax),%eax; %eax = x->f[1]
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 4(%eax),%eax
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->c;
}
```

```
int proc2(struct s2 *x) {
return x->f[1];
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax; %eax = x->c
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 12(%eax),%eax; %eax = x->f[1]
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 4(%eax),%eax; %eax = x->b
  movl 20(%eax),%eax
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->c;
}
```

```
int proc2(struct s2 *x) {
return x->f[1];
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax; %eax = x->c
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 12(%eax),%eax; %eax = x->f[1]
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 4(%eax),%eax; %eax = x->b
  movl 20(%eax),%eax; %eax = x->b.i->f[3]
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->c;
}
```

```
int proc2(struct s2 *x) {
return x->f[1];
}
```

```
int proc3(struct s1 *x) {
return x->____________________;
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax; %eax = x->c
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 12(%eax),%eax; %eax = x->f[1]
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 4(%eax),%eax; %eax = x->b
  movl 20(%eax),%eax; %eax = x->b.i->f[3]
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->c;
}
```

```
int proc2(struct s2 *x) {
return x->f[1];
}
```

```
int proc3(struct s1 *x) {
return x->b.i->f[3];
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax; %eax = x->c
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 12(%eax),%eax; %eax = x->f[1]
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 4(%eax),%eax; %eax = x->b
  movl 20(%eax),%eax; %eax = x->b.i->f[3]
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax; x -> %eax
  movl  (%eax),%eax
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->c;
}
```

```
int proc2(struct s2 *x) {
return x->f[1];
}
```

```
int proc3(struct s1 *x) {
return x->b.i->f[3];
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax; %eax = x->c
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 12(%eax),%eax; %eax = x->f[1]
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 4(%eax),%eax; %eax = x->b
  movl 20(%eax),%eax; %eax = x->b.i->f[3]
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax; x -> %eax
  movl  (%eax),%eax; %eax = x->_
  movl  24(%eax),%eax
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->c;
}
```

```
int proc2(struct s2 *x) {
return x->f[1];
}
```

```
int proc3(struct s1 *x) {
return x->b.i->f[3];
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax; %eax = x->c
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 12(%eax),%eax; %eax = x->f[1]
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 4(%eax),%eax; %eax = x->b
  movl 20(%eax),%eax; %eax = x->b.i->f[3]
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax; x -> %eax
  movl  (%eax),%eax; %eax = x->_
  movl  24(%eax),%eax; %eax = x->i->g
  movl  (%eax),%eax
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->c;
}
```

```
int proc2(struct s2 *x) {
return x->f[1];
}
```

```
int proc3(struct s1 *x) {
return x->b.i->f[3];
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax; %eax = x->c
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 12(%eax),%eax; %eax = x->f[1]
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 4(%eax),%eax; %eax = x->b
  movl 20(%eax),%eax; %eax = x->b.i->f[3]
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax; x -> %eax
  movl  (%eax),%eax; %eax = x->_
  movl  24(%eax),%eax; %eax = x->i->g
  movl  (%eax),%eax; %eax = x->i->g->d
  movsbl 1(%eax),%eax
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->c;
}
```

```
int proc2(struct s2 *x) {
return x->f[1];
}
```

```
int proc3(struct s1 *x) {
return x->b.i->f[3];
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax; %eax = x->c
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 12(%eax),%eax; %eax = x->f[1]
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 4(%eax),%eax; %eax = x->b
  movl 20(%eax),%eax; %eax = x->b.i->f[3]
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax; x -> %eax
  movl  (%eax),%eax; %eax = x->_
  movl  24(%eax),%eax; %eax = x->i->g
  movl  (%eax),%eax; %eax = x->i->g->d
  movsbl 1(%eax),%eax; %eax = x->i->g->d->a[1]
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->c;
}
```

```
int proc2(struct s2 *x) {
return x->f[1];
}
```

```
int proc3(struct s1 *x) {
return x->b.i->f[3];
}
```

```
char proc4(union u1 *x) {
return x->____________________;
}
```

Problem 7

```
struct s1 {
  char a[3]; 0
  union u1 b; 4
  int c; 8
};

struct s2 {
  struct s1 *d; 0
  char e; 4
  int f[4]; 8
  struct s2 *g; 24
};

union u1 {
  struct s1 *h; 0
  struct s2 *i; 0
  char j; 0
};
```

```
(a)proc1:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 8(%eax),%eax; %eax = x->c
  movl %ebp,%esp
  popl %ebp
  ret
(b)proc2:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 12(%eax),%eax; %eax = x->f[1]
  movl %ebp,%esp
  popl %ebp
  ret

(c)proc3:
  pushl %ebp
  movl %esp,%ebp
  movl 8(%ebp),%eax; x -> %eax
  movl 4(%eax),%eax; %eax = x->b
  movl 20(%eax),%eax; %eax = x->b.i->f[3]
  movl %ebp,%esp
  popl %ebp
  ret

(d)proc4:
  pushl %ebp
  movl  %esp,%ebp
  movl  8(%ebp),%eax; x -> %eax
  movl  (%eax),%eax; %eax = x->_
  movl  24(%eax),%eax; %eax = x->i->g
  movl  (%eax),%eax; %eax = x->i->g->d
  movsbl 1(%eax),%eax; %eax = x->i->g->d->a[1]
  movl  %ebp,%esp
  popl  %ebp
  ret
```

```
int proc1(struct s1 *x) {
return x->c;
}
```

```
int proc2(struct s2 *x) {
return x->f[1];
}
```

```
int proc3(struct s1 *x) {
return x->b.i->f[3];
}
```

```
char proc4(union u1 *x) {
return x->i->g->d->a[1];
}
```