Bresia Prudente, bprude2
CS 261

HW 4B

3.56

A.

Values	Registers
x	%esi
n	%ebx
result	%edi
mask	%edx

B. result = -1
mask = 1

C. mask != 0

D. The mask gets updated by "sall" at line 10

E. The result gets updated by "xorl" at line 8

F.
```
1 int loop(int x, int n)
2 {
3         int result = -1 ;
4         int mask;
5         for (mask = 1; mask != 0; mask = (mask << n)) {
6                 result ^= mask & x;
7         }
8          return result;
9 }
```

3.60

A.

x_a is the start of the array
T is the total number of height elements in the sub arrays or depth elements in each row
L is the size of T in bytes
C is the total columns in the sub arrays or total columns in each row
i is the index of a specific row of the element that we're trying to access
j is the index of the specific column of the element that we're trying to access
k is the index of the specific depth of the element that we're trying to access

With these values, we create the formula:

&A[i][j][k] = x_a + L((C*I + j)(T + k)

B.

Steps	Formula	Values	Registry (which values are stored in)
1	N/A	j	%edx
2	4 * j + j	5j	%eax
3	j + 2(5j)	11j	%eax
4	99 * i	99i	%edx
5	11j + 99i	11j + 99i	%eax
6	11j + 99i + k	11j + 99i + k	%eax
7	A[99i][11j][k]	A[99i][11j][k]	%edx
8	j+2	j+2	%eax
9	A[99i][11j][k]	A[99i][11j][k]	%eax
10	N/A	1980	%eax