**Assignment#2**

**Tianhao Liu**

**liux3271@umn.edu**

**Section: 014 (2:30pm on Thursday)**

**Problem #1**

unsigned prob2(unsigned n)

{

unsigned result = (\_\_\_1\_\_\_);

if (\_\_n == 0\_\_) return (\_\_1\_\_);

for (\_\_\_\_\_\_n\_\_\_\_\_\_; \_\_\_n!=0\_\_\_; \_\_\_n--\_\_\_)

{

\_\_\_result \*= n\_\_\_\_;

}

return \_\_\_\_\_result\_\_\_\_\_;

}

**Problem #2**

void hw-switcher (long x, long y, long z, long \*result)

{

long temp;

switch (z) {

case \_\_\_1\_\_\_: /\* Case A \*/

temp = \_\_\_\_\_\_-20+y+y\_\_\_\_\_\_\_\_\_\_\_;

break;

case \_\_\_0\_\_\_: /\* Case B \*/

y = \_\_\_\_\_\_\_\_\_x+12\_\_\_\_\_\_\_\_\_\_;

/\* fall through \*/

case \_\_\_3\_\_\_: /\* Case C \*/

temp = \_\_\_\_\_x+2y\_\_\_\_\_\_\_\_\_;

break;

case \_\_\_4\_\_\_: /\* Case D \*/

case \_\_\_5\_\_\_: /\* Case E \*/

temp = \_\_\_\_\_(x-y)<<4\_\_\_\_\_\_\_\_\_\_;

break;

default:

temp = \_\_\_\_\_\_\_x\_\_\_\_\_\_\_\_\_\_\_;

}

\*result = temp;

}

**Problem #3**

A.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| C0 | C1 |  | | | | | | p | | | | | | | | j | | | | i | |  | | a | | | | | | | |

B. 32

C. 8

D. 0

E. 8

F. 8

**Problem #4**

A. rdx

B. rax

C. 15

**Problem #5**

A.

|  |  |
| --- | --- |
| rsp+120 |  |
| rsp+112 |  |
| rsp+104 |  |
| rsp+96 |  |
| rsp+88 |  |
| rsp+80 |  |
| rsp+72 |  |
| rsp+64 |  |
| rsp+46 |  |
| rsp+48 |  |
| rsp+40 | h |
| rsp+32 |  |
| rsp+24 | &h |
| rsp+16 | g |
| rsp+8 | f |
| rsp+0 | e |

B. It takes a.y[0] from ($rsp+8), a.y[1] from ($rsp+16), a.y[2] from ($rsp+24). For a.z, it takes the address stored in ($rsp+32), which is ($rsp+24) for function evalsum since “call” will subtract rsp by 8, and take the value from ($rsp+40) that stores h.

C. The result is stored starting at ($rsp+80), it puts a.y[0] to ($rsp+80), a.y[1] from ($rsp+88), a.y[2] from ($rsp+96), a.z to ($rsp+104).

D.

|  |  |
| --- | --- |
| rsp+120 |  |
| rsp+112 |  |
| rsp+104 | h |
| rsp+96 | g |
| rsp+88 | f |
| rsp+80 | e |
| rsp+72 |  |
| rsp+64 |  |
| rsp+46 |  |
| rsp+48 |  |
| rsp+40 | h |
| rsp+32 |  |
| rsp+24 | &h |
| rsp+16 | g |
| rsp+8 | f |
| rsp+0 | e |

In the function change, rax takes an address from rdi at the very beginning which points to ($rsp+80) setting by evalsum before calling it. We put all elements to that address and its following places. When we return from change, rax stores the address of the memory where stores q. Since evalsum knows where that address is, it seeks values from ($rsp+80) directly and calculate the sum and puts the sum to rax to return.