Problem 1:

unsigned prob2(unsigned n) {

unsigned result = **1**;

if (**n == 0**) return **result**;

for (**n**; **n>0**; **n=n-1**) {

result = **result \* n** ;}

return **result**;

}

Problem 2:

void hw-switcher (long x, long y, long z, long \*result) //x in rdi, y in rsi, z in rdx, result in rcx

{

long temp;

switch (z) {

case 0: /\* Case A \*/ L3

temp = **x + 12**;

break;

case 1: /\* Case B \*/ L5

y = **2y - 20**;

/\* fall through \*/

case 3: /\* Case C \*/ L6

temp = **2y + x**;

break;

case 4: /\* Case D \*/L7

case 5: /\* Case E \*/L7

temp = **(x-y)<<4**;

break;

default: L2

temp = **result**;

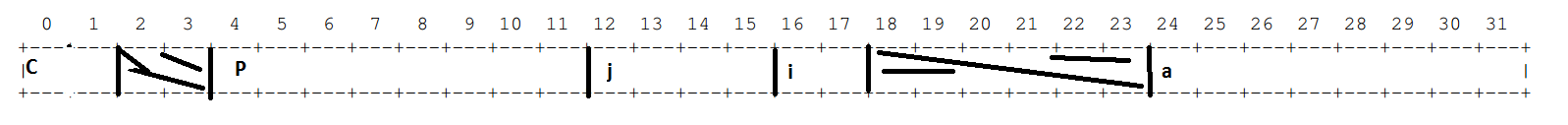
}

\*result = temp;

}

Problem 3:

a)



b) 32 bytes

c) 8 byte

d) pointer (8) + long (8) + int (4) + char[2] (2) + short (2) = 24

would be 8 unused, allocated bytes in the best case.

e) 8

f) 8

Problem 4:

a) rdx

b) rax

c) 15

Problem 5: e = rdi, f = rsi, g = rdx, h = rcx

a) strX on stack first 40 bytes ([8] pointer to strX, [24] array x, [8] \*pointer)

%rdi on stack, first argument

%rsi on stack, 2nd argument

%rdx on stack, 3rd argument

Stack: e, f, g, h

b) the function change accesses the elements of structures argument t using their offset from register %rsp.

c) it uses the register %rcx to move the elements of the array in structure t to array in structure a.