1. **[2+2 points]** What is link time error ? Please provide an example for such an error that can happen.

2. **[2+2 points]** Please provide two examples of abstractions provided by Operating System. Please explain (in brief) how each of them helps to abstract along with the component that gets abstracted.

3. Please perform the following Hex operations and present the output in Hex (please assume word size is not a limitation):

a) **[2 points]** 0x2BA + 0x04 = 0x2BE (in Hex)

b) **[2 points]** 0xFF - 0x1A = 0xE5 (in Hex)

(4)

c) **[2 points]** 0x00A * 0x26 = 0x190 (in Hex)

2(16+6 — 38 × 10 — 380

16	380	
16	23	
	1	

4. **[4 points]** Please convert 0.725 in decimal system into (unsigned) binary number. Please show all steps of the computation [and use the bits needed].

+ 0.450 | $(0)_{10} \rightarrow (0)_2$

5. **[4 points]** If you want to compute x*20 using assembly programming for a two's complement binary number x, what is the minimum number of shift operations plus additions plus subtractions needed to perform this computation ? Please show all the steps involved and mention whether the shift used is an Arithmetic or Logical shift (you do not need to write program) ?

6. Please answer the following regarding the T2U operator:

a) **[1+3 points]** What is T2U operator ? Please present the set of steps for how it can be performed using 8 bits for the decimal value -36 ?

T2U operator converts a 2's complement-

b) **[4 points]** Please explain using a C code snippet for how T2U (i.e., the equivalent implicit casting for it in C) can result in casting surprises ? Please present the C code snippet and brief explanation for each of the lines.

c) **[3 points]** How many integer registers are present in the X86-64 architecture ? Please provide names of any 4 registers (only first four registers will be considered if more are presented) ?

d) **[2 points]** What are the different operand combinations for the movq instruction (wrong combinations will receive **-0.5 points** each) ?

e) **[2+2 points]** What is the most general form of Memory Addressing Mode ?
Please mention what each of the terms stand for ?

Memory addressing mo

7. Please answer the following questions:

a) **[2 points]** Please provide names of two AMD X86 64 bit processors (only first two names will be considered) ?

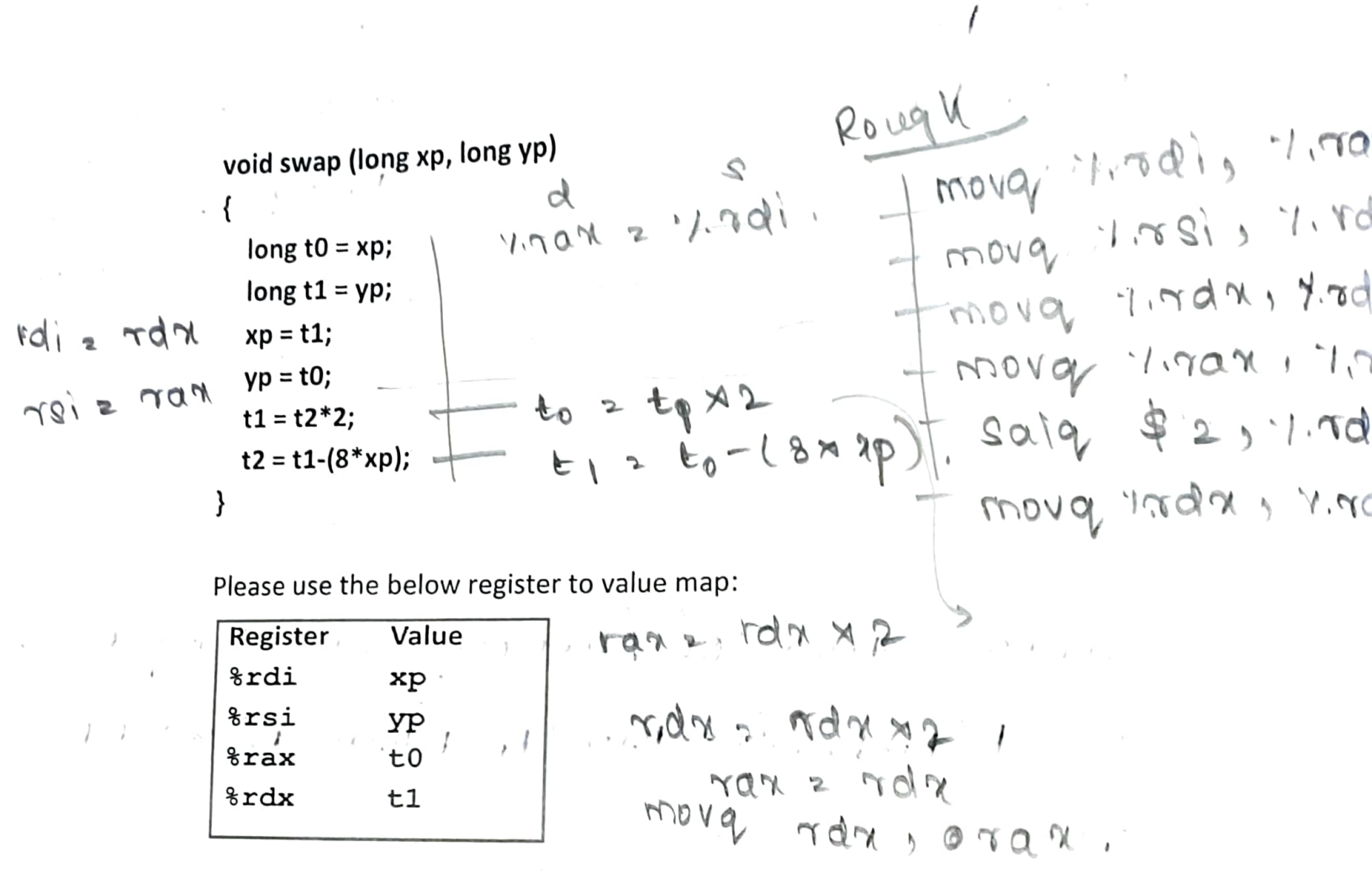
b) **[2 + 1 points]** Please explain the difference between Address space vs. Memory space ? Which would be a bigger space ?

8. **[6 points]** Please present assembly code for the following code snippet - it is sufficient to present code for the six lines within the swap function.

```
void swap (long xp, long yp)
{
  long t0 = xp;
  long t1 = yp;
  xp = t1;
  yp = t0;
  t1 = t2*2;
  t2 = t1-(8*xp);
}
```

Please use the below register to value map:

Register	Value
%rdi	xp
%rsi	yp
%rax	t0
%rdx	t1



9. Please answer the following questions – wrong answer has **-0.5** marks penalty:

a) **[1.5 points]** Suppose register %rdi holds value p and %rsi holds value q.
For leaq 4(, %rdi, 2), %rax Result = 2p+4

b) **[1.5 points]** Suppose that a and b have byte values 0x213 and 0x54, respectively. a & b = 0x010

c) **[1.5 points]** Suppose that a and b have byte values 0x01 and 0x1, respectively. a | ~b = 0xFF

d) **[1.5 points]** Suppose that a and b have byte values 0x12 and 0x21, respectively. a && b = 0x01

e) **[1 points]** shrq $2, $0x012 = 0x004

f) **[1 points]** mov**b** $0xFF, %bl
[Example] mov__l_ %eax, %dx [can skip zero or sign extension]

g) **[2 points]** Suppose register %rdx holds z, instruction leaq (%rdx, %rdx, 2), %rax followed by salq $4, %rax = 4*8*z