

Problem 1: Symbol Table (14points)

1. [1]	Y	[2]	OBJECT	[3]	00000002	[4]	2	
	[5]	COM	[6]	GLOBAL	[7]	12	[8]	4
	[9]	Y	[10]	12	[11]	3	[12]	Y
	[13]	NOTYPE	[14]	00000000	[15]	0	[16]	UND
	[17]	Y	[18]	FUNC	[19]	000000000	[20]	1

2. 6 3 8 5

Problem 2: PIC (8points)

1. [1] * 0x804a01c [2] 0x08048356

2. 0x08049f14

3. the real address of function printf.

Problem 3: Linking (20points)

1. [1]	0x8	[2]	fc ff ff ff	
	[3]	(%eax)	[4]	04 00 00 00
2. [5]	00000008	[6]	R_386_PC32	
	[7]	00000000	[8]	a
	[9]	00000004	[10]	R_386_32
	[11]	fun1	[12]	00000004
3. [13]	08040a38			
	[14]	0804a020		
	[15]	08048414		
	[16]	080483f3:	c7 05 40 a0 04 08 08 00 00 00 movl \$0x8,0x804a040	
	[17]	080483fd:	e8 12 00 00 00 call 8048414 <fun2>	
	[18]	08048407:	a1 1c a0 04 08 mov 0x804a01c,%eax	
	[19]	08048417:	e8 e8 ff ff ff call 8048404 <fun1>	
	[20]	0804a030:	24 a0 04 08	

Problem 4: HCL (8points)

1. `bool xor = (!a && b) || (a && !b);`

```
2. int Out = [
    !s2 && !s1 && !s0 : A; # 000
    !s2 && !s1 && s0  : B; # 001
    !s2 && s1 && !s0   : C; # 010
    !s2 && s1 && s0    : D; # 011
    !s1 && !s0         : E; # 100
    !s1               : F; # 101
    !s0               : G; # 110
    1                 : H; # 111
];
```

```
3. int Max4 = [
    A>=B && A>=C && A>=D : A;
    B>=C && B>=D :      B;
    C>=D :          C;
    1:             D;
];
```

Problem 5: Y86 (15points)

- | | | |
|-----------------------------------|-----------|-----------------------------|
| 1. [1] 8024000000 | [2] 0x014 | [3] 0x014 |
| [4] 263 | [5] 0x024 | [6] <code>pushl %ebx</code> |
| [7] 30f214000000 | [8] 0x048 | [9] 6003 |
| [10] <code>subl %ebx, %ecx</code> | | |
| 2. [1] 0xffffffff334 | [2] 0x600 | |
| [3] 0 | [4] 1 | [5] 0 |

Problem 6: Processor (35points)

1.

Field	retxx
Fetch	<code>icode:ifun <- M1[PC]</code> <code>valP <- PC+1</code>
Decode	<code>valA <- R[%esp]</code> <code>valB <- R[%esp]</code>
Execute	<code>valE <- valB + 4</code> <code>Cnd <- Cond(CC, ifun)</code>
Memory	<code>valM <- M4[valA]</code>
Write Back	<code>if(Cnd)</code> <code>R[%esp] <- valE</code>
PC update	<code>PC <- Cnd ? valM : valP</code>

2.

Condition	Trigger
condition ret	<code>IRETXX in {E_icode, M_icode} && Cnd</code>

Condition	Pipeline register				
	F	D	E	M	W
condition ret	S	B	B	-	-

3.

M	
E	retxx
D	

M	retxx
E	B
D	B

M	
E	
D	ret

M	
E	ret
D	B

M	ret
E	B
D	B

Condition	Pipeline register				
	F	D	E	M	W
condition ret	S	B	B	-	-
ret	S	B	-	-	-
combination	S	B	B	-	-

4. [1] 3 [2] 0.04 [3] 0.3
 [4] 0.03 [5] 0.015 [6] 1.385
 5. [1] 28 [2] 9 [3] 26 [4] 3