Seminar 1

1. $-49_{10} = 11001111_2$

sign extended: $1111111001111_2 = 0xFCF$

zero extended: $0000110011111_2 = 0x0CF$

$113_{10} = 01110001_2$

sign extended: $0000011100001_2 = 0x071$

zero extended: $0000011100001_2 = 0x071$

2. Cleaning up z:

z & = 1111 1111 1111 1111 1111 1111 0000 0000

***

getting least significant 3 bits from y and shifting 5 positions:

((y & 0000 0000 0000 0000 0000 0000 0000 0111) << 5)

*

extracting bits from x and shifting so that last significant starts at position 0.

((x & 0000 0000 0000 0011 1110 0000 0000 0000) >> 13)

**

* = 0x7 , ** = 0x3E000 , *** = 0xFFFFFF00

all in one statement:

z = (z & 0xFFFFFF00) | ((y & 0x3) << 5) | ((x & 0x3E000) >> 13);

3.

```
# include <stdio.h>

void adder(const int *x, const int *y, int *z) {

    *z = *x + *y;

}

int foo(int a) {

    const int k = 10;
    int p;
    adder(&k, &a, &p);

    return p;

}
```

4. machine code: $0 \times 2d28fff9$

$= 0010\ 1011\ 0010\ 1000\ 1111\ 1111\ 1111\ 1001_2$

opcode $= 001010_2 = 10_{10}$ => slti

slti is an I-type instruction

rs $= 11001_2 = 25$ => $t9

rt $= 01000_2 = 8$ => $t0

imm $= 1111\ 1111\ 1111\ 1001_2 = 65529$

The MIPS instruction is: slti $t0, $t9, 65529

5. J-Type instruction, 6 bits is reserved for the opcode and 26 bits for the target address

Opcode: 000010

The 32-bit jump adress is:

0000 0000 0100 0000 0000 0000 0010 1100

The 26-bit target adress is:

0000 0100 0000 0000 0000 0010 11

The machine code becomes:

000010 0000 0100 0000 0000 0000 0010 11

6.

```
void square_reverse(const double *x, double *y, int len) {
    int i;
    for (i=0; i<len; i++) {
        y[i] = x[len-1-i]
    }
}
```