

There are two question for Quiz 1 Theory Section Aj. Prabhas

1) Control flow

This is a C function. Translate the following "switch" statement into assembly code of P1. (defined below)

```
int myfunc(int x){
    int a;

    a = 0;
    switch(x){
        case 11: a = 1; break;
        case 22: a = 2; break;
        case 33: a = 3; break;
    }
    return a;
}
```

Assume 32-bit "int". assume local variables x, a are in registers.

P1 is a 32-bit processor with 32 register, r0-r31. The instruction set of P1 (to be used in this question) is as follows:

```
mov r1, r2          ; move data from r2 to r1
mvi r1, n            ; move n to r1
beq r1, r2, label    ; branch if r1 == r2 to label
bne r1, r2, label    ; branch if r1 != r2 to label
jmp label            ; unconditional jump to label
```

2) Scope and binding

Here is code in Python. f1() creates a list of 10 elements.

What is the screen output when main1() is executed?

Now, try to execute main2(). Why the result is different?

Please explain this difference using the theory of scope and binding.

```
def f1():
    A = []
    for i in range(0,10):
        A.append(i)
    return A
```

```
def main1():
    a = f1()
    print(a)
    b = a
    print(b)
    a[2] = 88
    print(b)

def main2():
    a = f1()
    print(a)
    b = list(a)
    print(b)
    a[2] = 99
    print(b)
```