```
# ECE 391 Problem Set 1, Sep 2nd

# NetID: yuwang17
# Name: Yu Wang


# 1. Mapping C to Assembly

.GLOBAL dispatch
dispatch:
    pushl %ebp                  # save old frame pointer
    movl %esp, %ebp             # point to new frame
    movl $1, %edx               # edx = 1, int bit = 1
    xorl %ecx, %ecx             # ecx = 0
loop:
    cmpl $32, %ecx              # compare ecx and 32
    jge return_0                # if ecx < edi, jump to return 0
    testl 8(%ebp), %edx          # if bitmask & bit
    jz continue
    jmp *jump_table(, %ecx, 4)  # return (jump_table[i])(arg)
continue:
    shll $1, %edx               # edx = edx << 1
    incl %ecx                   # increment ecx
    jmp loop                    # back to loop
return_0:
    movl $0, %eax               # eax = 0
    leave                       # restore frame pointer and stack
pointer
    ret                         # return


/* 2. Understanding Disassembled Functions */

int calculate(unsigned int arg1, unsigned int arg2) {
    switch (arg1) {
        case 0: return arg2 * arg2;
        case 1: return -arg2;
        case 2: return arg2 + 0x80;
        default: return arg2;
    }
}
```