

```
int x, y;
int result;
```

```
int main(int argc, char *argv[])
{
    if (argc != 3) { fprintf(stderr, "usage: absdiff <num> <num>\n"); exit(1); }
    x = atoi(argv[1]);
    y = atoi(argv[2]);
```

```
if (x > y) {
    result = x - y;
} else {
    result = y - x;
}
```

```
80484fa: mov  %eax,%edx           # y in register %edx
80484fc: mov  %eax,0x80497c8      # y's memory location
8048501: mov  0x80497c4,%eax      # x in register %eax
8048506: cmp  %eax,%edx           # cmp x,y
8048508: jge  8048513             #
804850a: sub  %edx,%eax           #
804850c: mov  %eax,0x80497c0      #
8048511: jmp  0x804851b           #
8048513: sub  %eax,%edx           #
8048515: mov  %edx,0x80497c0      # result (0x80497c0)
804851b: ...
```

```
printf("result: %d\n", result);
```

```
return result;
```

```
}
```

CHALLENGE

Trace through the code when:

1) x = 10, y = 100

2) x = 100, y = 10

What is the sequence of PC (%eip) values that occur?

What value is in the registers at each step?

Other questions:

What are the different jump instructions that are used?

What are the CC values after the cmp instruction?

How would you rewrite to use "jump less than" (jl) instead of jge?

