预习作业 1

```
/* LOOP #1 */
for (i = 0; i < N; i++) {
    a[i] = a[i] * 2000;
    a[i] = a[i] / 10000;
}

/* LOOP #2 */
b = a;
for (i = 0; i < N; i++) {
    *b = *b * 2000;
    *b = *b / 10000;
    b++;
}
```

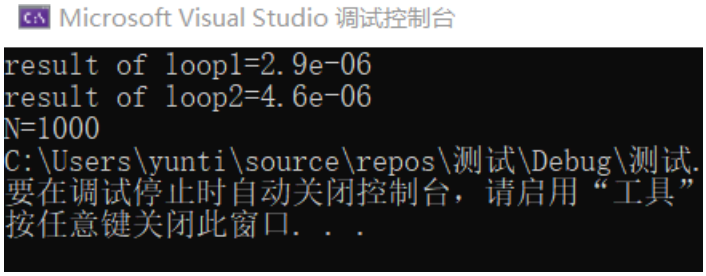
为了获得运行更快的目标程序，选择哪种编程方式?

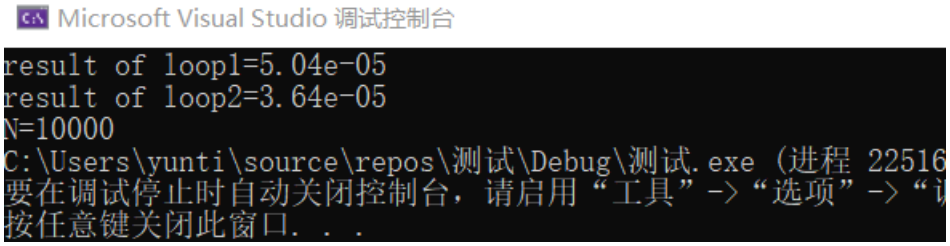
用 QueryPerformance 函数计算运行时间

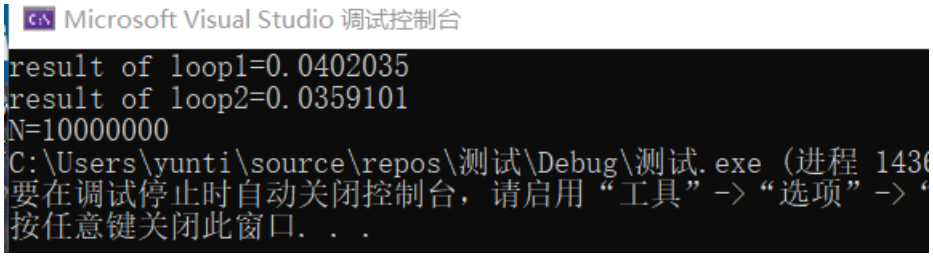
```
int main() {
    int N = 10000000;
    int* a = new int[N], * b = new int[N];
    for (int i = 0; i < N; i++) {
        a[i] = b[i] = (i + 3) * 7;
    }

    LARGE_INTEGER freq, start1, end1;
    QueryPerformanceFrequency(&freq);
    QueryPerformanceCounter(&start1);

    for (int i = 0; i < N; i++) {
        a[i] = a[i] * 2000;
        a[i] = a[i] / 10000;
    }

    QueryPerformanceCounter(&end1);
    std::cout << "result of loop1=" << (end1.QuadPart - start1.QuadPart) * 1.0 / freq.QuadPart << std::endl;

    LARGE_INTEGER start2, end2;
    QueryPerformanceCounter(&start2);

    for (int i = 0; i < N; i++) {
        *b = *b * 2000;
        *b = *b / 10000;
        b++;
    }
```

```
    QueryPerformanceCounter(&end2);
    std::cout << "result of loop2=" << (end2.QuadPart - start2.QuadPart) * 1.0 / freq.QuadPart << std::endl;
    std::cout << "N=" << N;
}
```

获得结果如下







单位为秒

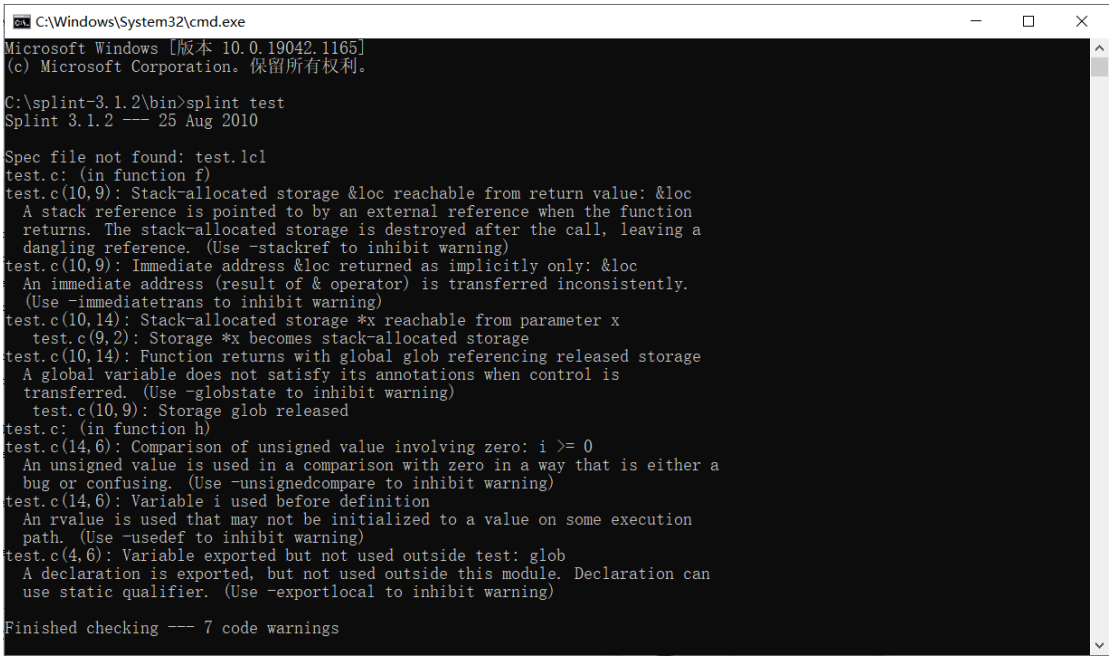
即使在 N 相同的情况下重复试验，两种循环消耗时间结果也互有长短。但通过修改 N 的值、编译优化级别，或者直接把代码复制到其他机器运行，结果应该会有不同。

预习作业 2

```
char firstChar1(char* s) {
        return *s;
}
int* glob;
int* f(int** x) {
        int sa[2] = { 0,1 };
        int loc = 3;
        glob = &loc;
        *x = &sa[0];
        return &loc;
}
void h(void) {
        unsigned int i;
        if (i >= 0)
                printf(">=0\n");
        else
                printf("<0");
}
```

这段程序有什么错误?

根据 splint 检查的结果，有 7 个 code warning



函数 f 中:

&loc 分配的堆栈内存能通过返回值到达；&loc 直接地址返回为隐式的；*x 分配的堆栈内存能通过变量到达；函数返回全局引用已释放的内存

函数 h 中:

Unsigned 值用来和 0 比较；i 使用前未声明；glob 变量导出但并未在外部使用

实际使用 vs 编译只显示 1error 和 2warning，所有警告级别输出都一样

预习难点

0：本学期没有选上汇编语言课，担心不能跟上实验要求

1：怎么创建符号表能够使得所有合法的语句都能够流畅自然的分解重组成能够分析的语法树，而不至于在某一个分析阶段卡住或生成歧义

怎么在扫描间确定变量对应的机器码中地址