1. 按照第4部分（软件测试基本过程）介绍的简单测试流程，构造下面程序段的测试流程 。   
void static main (int argc, char\* argv[])   
{   
    int x = atoi ( argv[1] );   
    int y = atoi ( argv[3] );   
    switch ( \*argv[2] ) {   
      case ‘+’: /// 两个数相加   
        printf( “%d + %d  = %d”, x, y, x+y); break;   
      case ‘-’: /// 两个数相减   
        printf( “%d - %d  = %d”, x, y, x-y); break;   
      case ‘\*’: /// 两个数相乘   
        printf( “%d \* %d  = %d”, x, y, x\*y); break;   
      case ‘/’: /// 两个数相除   
        printf( “%d / %d  = %d”, x, y, x/y); break;   
      default: /// 输入其它符号打印正确的使用方法   
        printf( “Correct use: x +(-\*/) y ”); break;   
      }   
}

1. 测试计划：黑盒测试
2. 测试设计：运行程序、查看结果
3. 建立测试用例：
4. 执行测试用例
5. 记录运行结果并分析（C~E的表格，假设整数的加减乘除如下:）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 用例编号 | 说明 | 操作过程 | 输入值 | 期望结果 | 实际输出 | 结果 |
| 1 | 测试加法功能 | 运行程序 | 1 + 1 | 1 + 1 = 2 | 1 + 1 = 2 | true |
| 2 | 测试乘法功能 | 运行程序 | 2 \* 3 | 2 \* 3 = 6 | 2 \* 3 = 6 | true |
| 3 | 测试除法功能 | 运行程序 | 3 / 2 | 3 / 2 = 1 | 3 / 2 = 1 | true |
| 3 / 0 | 抛出错误 | 程序终止 | false |
| 4 | 测试减法功能 | 运行程序 | 4 - 2 | 4 - 2 = 2 | 4 - 2 = 2 | true |
| 5 | 错误输入 | 运行程序 | 三个都不是数字 | 抛出错误 | 程序终止 | false |
| 6 | 两个不是数字 | 抛出错误 | 程序终止 | false |
| 7 | 一个符号但位置不对 | 抛出错误 | 程序终止 | false |

1. 1. 你认为软件测试和软件调试的区别是什么？
2. 软件测试是找到程序中的错误，调试是去找错误，然后修改。
3. 软件测试一般先于调试，即测试发现错误，然后调试定位并修改。
4. 软件测试确定的输入，确定的输出，软件调试不确定的输入，不确定的输出。
5. 软件测试可以很好的计划，而调试主要依靠程序员的嗅觉。
6. 软件测试不一定是开发人员，调试一定是开发人员。
7. 测试能够自动化、规模化，而调试必须依靠人慢慢来。
8. 测试的可以是文档或者代码，调试必须是代码。  
   2. 移动（手机）应用程序测试与普通应用程序测试相比，你认为其功能测试、性能测试有哪些不同的特点？（自由发挥）   
   (0.0分)
9. 用户界面的大小不同，app界面小，精简，PC的界面就复杂得多。
10. 兼容的操作系统不同，手机是mac和Android，PC是Windows、Mac、Linux。
11. 用户在使用上，APP呈现时间碎片化，PC的使用时间更为集中。
12. 手机作为通信设备，需要考虑电话、短信等消息的影响。
13. 使用方式的不同，APP靠点击屏幕，PC外设是鼠标和键盘。