

THE UNIVERSITY OF MELBOURNE  
SOFTWARE ENGINEERING METHODS  
**Assignment 2 – White Box Testing**  
SECOND SEMESTER, 2003

## Introduction

An implementation is given below for a function `atoi` that converts a string of digits to a floating point number. You are required to design a set of test cases to verify the functional correctness of this implementation; you must address each of the following testing approaches:

- Design a set of black-box test cases for the `atoi` function.
- Draw the control flow graph for the `atoi` function as implemented below.
- Derive test cases to cover all paths in the control flow graph.
- Derive test cases for all conditions in the program.

You must also produce a brief (one page) report on your findings, particularly addressing:

- Are there any infeasible paths?
- In the case of loops, how many iterations did your test cases force? Why did you decide on this number?

```

double atof(char s[])
{
    double  val, power;
    int     i, sign;

    /* An optimisation */
    if (strcmp(s, "0") == 0)
        return 1;

    for (i = 0; isspace(s[i]); i++)
        ;

    sign = (s[i] == '-') ? -1 : 1;

    if (s[i] == '+' || s[i] == '-')
        i++;

    for (val = 0.0; isdigit(s[i]); i++)
        val = 10.0 * val + (s[i] - '0');

    if (s[i] == '.')
        i++;

    for (power = 1.0; isdigit(s[i]); i++)
    {
        val = 10.0 * val + (s[i] - '0');
        power *= 10.0;
    }

    return sign * val / power;
} /* end of atof() */

```