

```

void fill_buffer ()                               //1 or En
{                                                  //2
    char *p;                                     //3
    int finished_a_line;                         //4
    if (bp_save != 0)                            //5
    {                                             //6
        buf_ptr = bp_save;                       //7
        buf_end = be_save;                       //8
        bp_save = be_save = 0;                   //9
        if (buf_ptr < buf_end)                   //10
            return;                              //11 or Ex
    }                                             //12
    if (*in_prog_pos == '\0')                    //13
    {                                             //14
        cur_line = buf_ptr = in_prog_pos;         //15
        had_eof = true;                           //16
        return;                                   //17
    }                                             //18
    p = cur_line = in_prog_pos;                  //19
    finished_a_line = 0;                          //20
    do                                           //21
    {                                           //22
        while (*p == ' ' || *p == TAB)            //23
            p++;                                  //24
        if (*p == '/' && (*(p + 1) == '*' || *(p + 1) == '/')) //25
        {                                       //26
            p += 2;                             //27
            while (*p == ' ' || *p == TAB)       //28
                p++;                             //29
            if (!strncmp (p, "INDENT-OFF", 12))    //30
                inhibited = 1;                   //31
        }                                       //32
        while (*p != '\0' && *p != EOL)           //33
            p++;                                 //34
        if (*p == EOL)                           //35
        {                                       //36
            finished_a_line = 1;                 //37
            in_prog_pos = p + 1;                 //38
        }                                       //39
        else if ((unsigned int) (p - current_input->data) < current_input->size) //40
        {                                       //41
            WARNING ("Warning: File %s contains NULL-characters\n", //42
                current_input->name, 0);         //43
            p++;                                 //44
        }                                       //45
        else                                   //46
        {                                       //47
            in_prog_pos = p;                     //48
            finished_a_line = 1;                 //49
        }                                       //50
    }                                           //51
    while (!finished_a_line);                   //52
    buf_ptr = cur_line;                         //53
    buf_end = in_prog_pos;                      //54
    if (buf_break && (buf_break->offset >= e_code - s_code || buf_break->offset <= 0)) //55
        clear_buf_break_list ();               //56
}                                              //57 or Ex

```

## CO7206

### Tutorial #2

Oct 14<sup>th</sup>, 2003

For the given C function,

1. draw the CFG, dominance tree, Post-dominance tree and the Control Dependence Graph.

\*Note: To simplify the graph, do not include all the steps, but only maximal blocks, excluding the branching statement, i.e. include only the statement whose numbers are in bold underlined font.

2. What is the post-dominators sets for statements 23, 34, 37

McCabe = 20  
 Halstead Program Diff = 22  
 Halstead Intel Content = 55  
 Halstead Mental Effort = 31457  
 Fan in = 4  
 Fan out = 3