

CSE 340 Spring 2016

HOMEWORK 5

Assigned 4/12/2016

Due Monday, 4/18/2016 by 11:59:59 pm

Remember that late submissions are not accepted for homeworks.

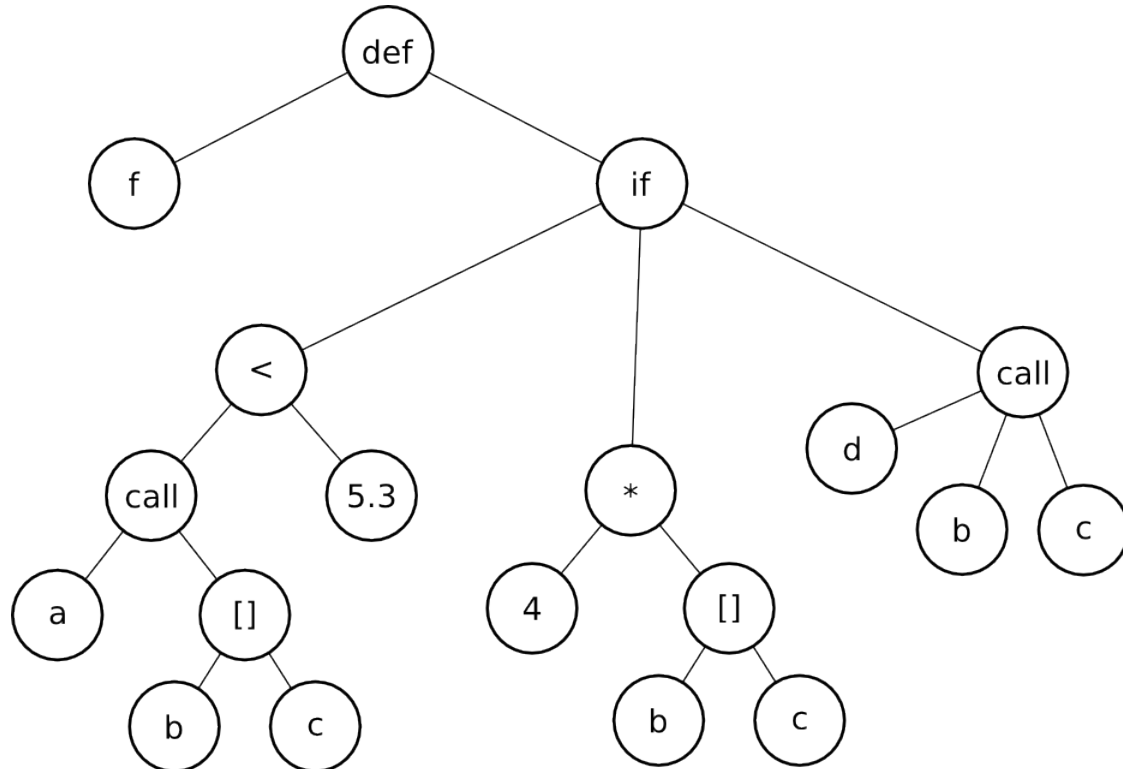
All submissions should be typed. Exception can only be made for drawing diagrams, which can be hand drawn and scanned in the submitted document.

You should show your work for all problems.

Problem 1. Consider the following definition:

```
fun f(a, b, c, d) = if a(b[c]) < 5.3 then
                    4 * b[c]
                    else
                    d(b, c)
```

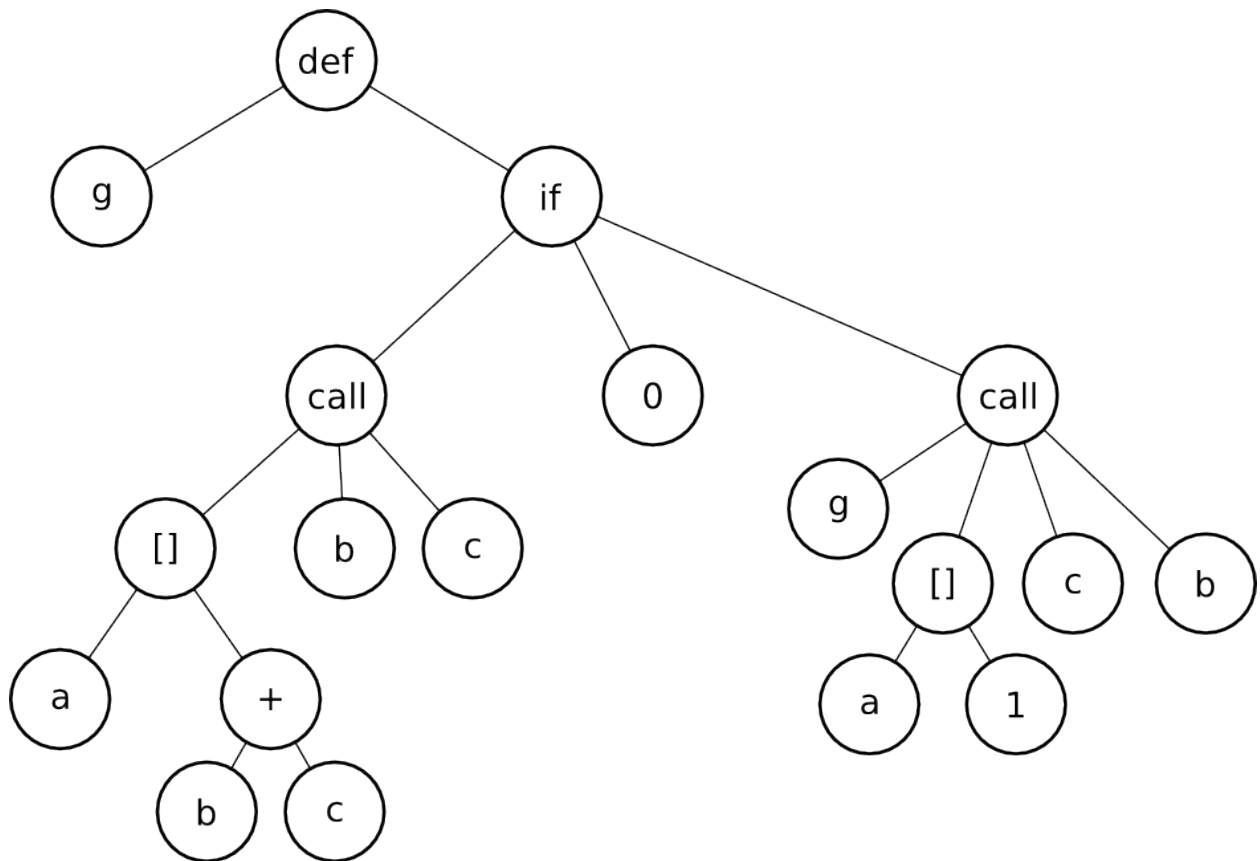
Using Hindley-Milner type inference, determine the type of f.



Problem 2. Consider the following definition:

```
fun g(a, b, c) = if a[b + c](b, c) then
  0
else
  g(a[1], c, b)
```

Using Hindley-Milner type inference, determine the type of g .



Problem 3. Consider the following code in C syntax:

```
int b;
int temp = 3;

int set_b(int i, int a)
{
    i = i + 1;
    b = a;
}

int p(int x, int S)
{
    for (x = 0; x < 3; x++) {
        temp = temp + S;
        S = temp;
    }
    return temp + b;
}

main()
{
    int a[3][3];
    int b;
    int r;
    int temp = 4;
    int i = 1;
    set_b(i, a[i][i]);
    a[0][0] = 100; a[0][1] = 10;  a[0][2] = 10;
    a[1][0] = 1;   a[1][1] = 10;  a[1][2] = 100;
    a[2][0] = 100; a[2][1] = 100; a[2][2] = 100;
    b = 0;
    r = p(i, a[i][b]);
    printf("%d %d %d %d\n", i, temp, b, r);
    printf("%d %d %d\n", a[0][0], a[0][1], a[0][2]);
    printf("%d %d %d\n", a[1][0], a[1][1], a[1][2]);
    printf("%d %d %d\n", a[2][0], a[2][1], a[2][2]);
}
```

- What is the output of the program if parameters are *passed by value*?
- What is the output of the program if parameters are *passed by reference*?
- What is the output of the program if parameters are *passed by name*?

Note: If during execution there is an attempt to access an element that is out of the array bounds, you should write *segmentation fault* for the output. Assume all variables are initialized to 0.

Problem 4. Consider the following program written in Ada syntax with the execution stack shown on the side. The line numbers are used to refer to the code and are not part of the code.

<pre> 01 procedure env is 02 x_env: integer; 03 04 procedure a is 05 x_a: integer; 06 procedure b is 07 x_b: integer; 08 procedure c is 09 x_c: integer; 10 begin 11 x_env = x_b + x_a; 12 a; 13 end c; 14 begin 15 x_b = x_a; 16 c; 17 end b; 18 19 procedure d is 20 begin 21 b; 22 end d; 23 begin 24 d; 25 end a; 26 begin 27 a; 28 end env; </pre>	<table border="0"> <tr><td>env</td><td> </td><td></td><td> </td><td>1</td></tr> <tr><td>a</td><td> </td><td></td><td> </td><td>2</td></tr> <tr><td>d</td><td> </td><td></td><td> </td><td>3</td></tr> <tr><td>b</td><td> </td><td></td><td> </td><td>4</td></tr> <tr><td>c</td><td> </td><td></td><td> </td><td>5</td></tr> <tr><td>a</td><td> </td><td></td><td> </td><td>6</td></tr> <tr><td>d</td><td> </td><td></td><td> </td><td>7</td></tr> <tr><td>b</td><td> </td><td></td><td> </td><td>8</td></tr> <tr><td>c</td><td> </td><td></td><td> </td><td>9</td></tr> <tr><td>a</td><td> </td><td></td><td> </td><td>10</td></tr> <tr><td>d</td><td> </td><td></td><td> </td><td>11</td></tr> <tr><td>b</td><td> </td><td></td><td> </td><td>12</td></tr> </table>	env				1	a				2	d				3	b				4	c				5	a				6	d				7	b				8	c				9	a				10	d				11	b				12
env				1																																																									
a				2																																																									
d				3																																																									
b				4																																																									
c				5																																																									
a				6																																																									
d				7																																																									
b				8																																																									
c				9																																																									
a				10																																																									
d				11																																																									
b				12																																																									

- Explain how the access links of activation records 6, 7, and 8 are found.
- Explain how the address of x_a in line 11 is calculated. Also, give the formula for the address.
- Explain how the address of x_a in line 15 is calculated. Also, give the formula for the address.