# Exercise Assignment 3

Brandon Hosley

Mike Davis

Exercise Assignment 3

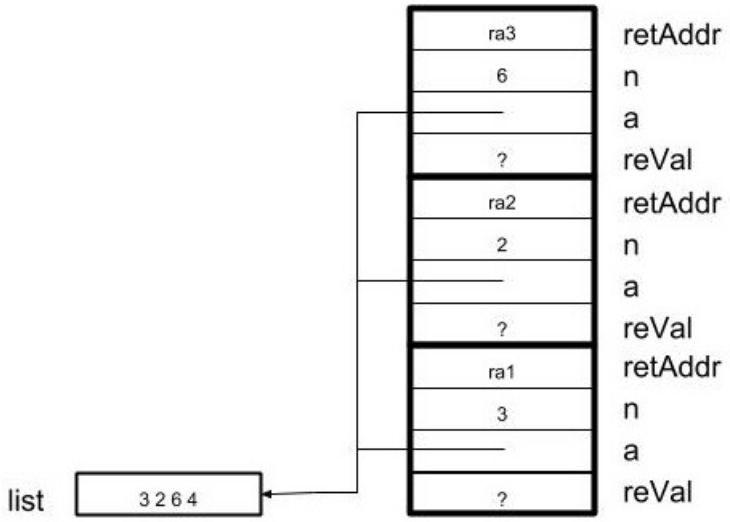
```
// Fig. 2.25 Transcribed:

#include <iostream>
using namespace std;

int list[4];

int sum (int a[], in n) {
    //returns the sum of the elements of a between a[0] and a[n].
    if (n == 0) {
        return a[0];
    }
    else {
        return a[n] + sum(a, n-1); //ra2
    }
}

int main () {
    cout << "Enter four integers: ";
    cin >> list[0] >> list[1] >> list[2] >> list[3];
    cout << "Their sum is: " << sum(list, 3) << endl; ra1
    return 0;
}

/*
Interactive Input/Output
Enter four integers: 3 2 6 4
Their sum is: 15
*/
```

### 2.1 a)

**How many times is sum called altogether?**
The sum function is called 4 times, three times it recurses and the fourth returns a[0].

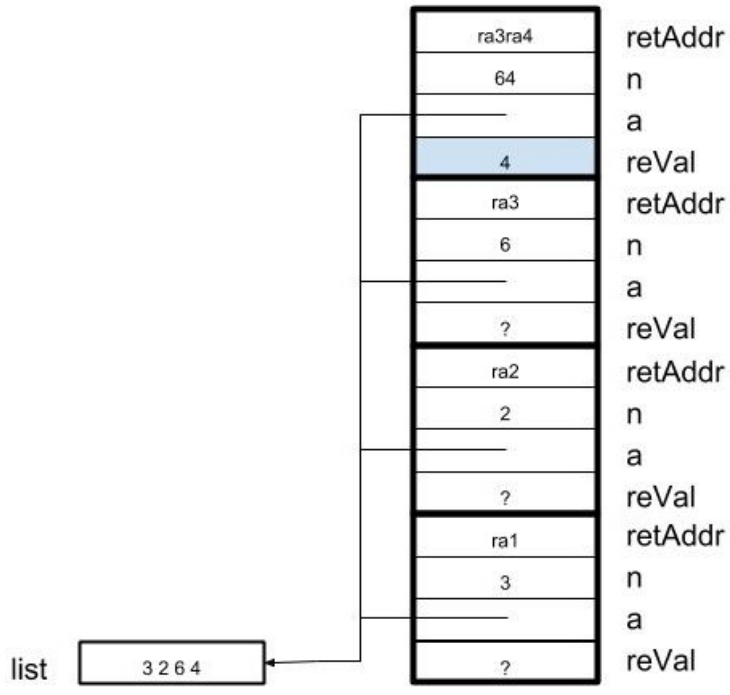
### 2.1 b)

**Draw a picture of the main program's variables and runtime stack at the time of the 3rd call ( should be 3 stack frames ).**



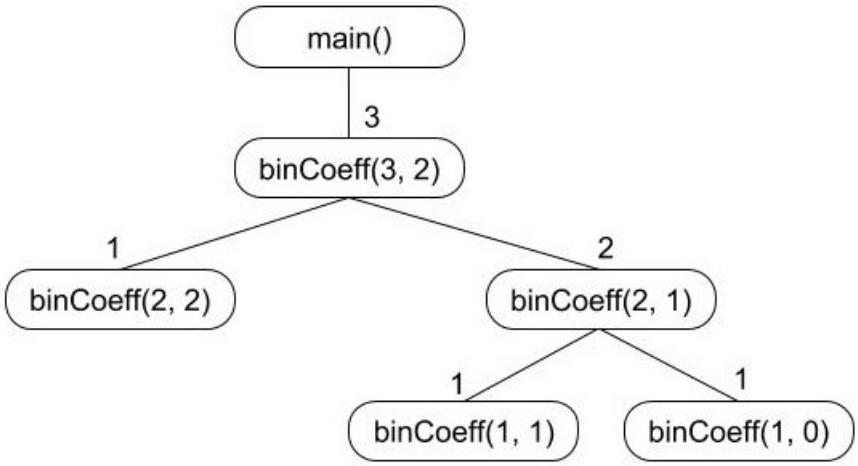

**2.1 c)**

**Draw the stack just before the return from the call in part (b).**




**2.2 c)**

**Draw the call tree of binCoeff(3,2) of fig 2.28 as seen in fig. 2.30**

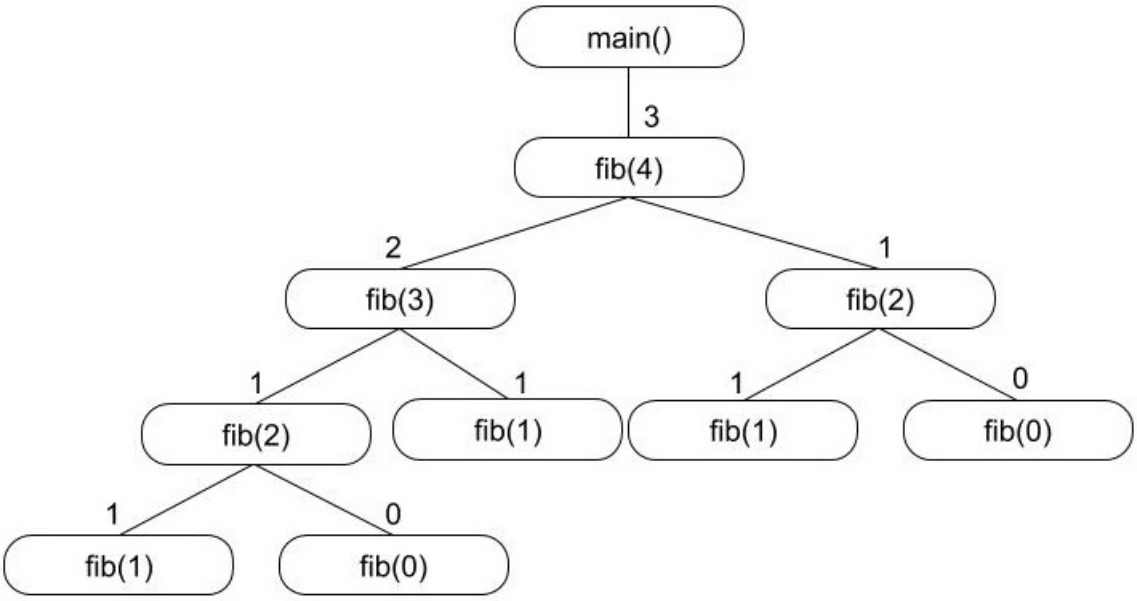



The function binCoeff() is called five times total including the initial binCoeff(3,2) This give a total of 6 stack frames with main() included.

The order of calls and returns are such that the left side of each pair is called and returned first. That is, binCoeff(2,2) before binCoeff(2,1) and binCoeff(1,1) before binCoeff(1,0).

### 2.5 b)

**Draw the call tree for *fib*(4)**




The function fib() is called 9 times including the initial fib(4). This means that there are a max number of 10 stack frames allocated at runtime, including the stack frame for main().

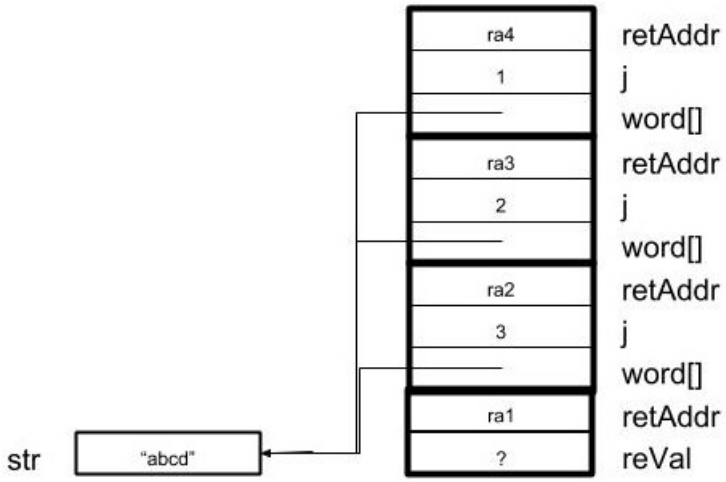
### 2.8 a)

**Examine the following C++ program:**

```
// Warford Book Example C++ program.

#include <iostream>
using namespace std;
```

```
void what (char word[], int j){
    if (j > 1){
        word[j] = word[3 - j];
        what(word, j - 1);
    } // ra2
}

int main() {
    char str[5] = "abcd";
    what(str, 3);
    cout << str << endl;
    // ra1
    return 0;
}
```

**Draw the run-time stack just after the first call.**




**2.8 b)**

**What is the output of the program?**
The program will output "abba".

References

Warford, J. (2009). *Computer systems* (4th ed.). Jones and Bartlett.