**Instructor:** Tadeusz S. Obuchowicz
**Date:** Friday, March 6, 2009
**Time Allowed:** 1 hour and 30 minutes

**Material Allowed:** Any C++ textbook, course notes. NO laptaps allowed. ENCS approved electronic calculators. Pencils, pens, erasers, rulers.

**Instructions:** Answer all 4 questions in the answer books provided. If you make any assumptions, clearly indicate so in your answer booklet.

**Question 1:** [25 points]

In a country far, far, away during a time long, long ago there lived a king called **Keith**. Keith ruled **Stonesland** and serenaded the people living in this idyllic land with his guitar playing. The currency used in Stonesland consisted of the following coins: a STONE, a KEITH, a MICK, and a RON (listed in DECREASING order of monetary value, refer to Figure 1).

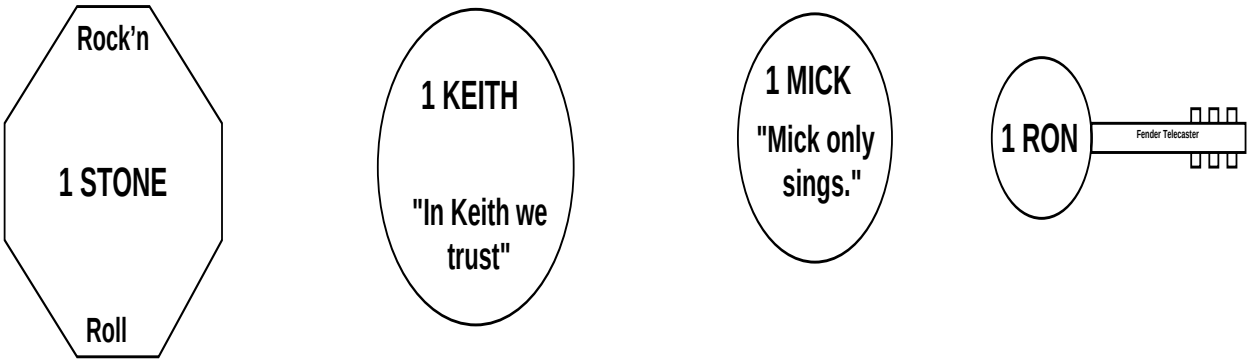


Figure 1: The coins used in Stonesland.

The relationship among these coins was decided by the King Keith (quite arbitrarily) to be:

1 STONE = 15 KEITHS
1 KEITH = 20 MICKS
1 MICK = 12 RONS
1 RON = 1 RON ( a RON is sort of like a penny, it is worth the least amount and people keep throwing them onto the streets of Stonesland much to the annoyance of the King ...)

Write a complete C++ program which will ask the user to input the number of STONE coins, the number of KEITH coins, the number of MICK coins, and the number of RON coins that she has in her pocket. The program will then display total value represented by these coins using the **LEAST amount** of coins. Typical interaction with the program would be similar to:

```
ted@brownsugar Programs 7:54pm >stones_currency
Enter the amount of STONES that you have 2
Enter the amount of KEITHS that you have 12
Enter the amount of MICKS that you have 68
Enter the amount RONS that you have 258

The amount you have entered is equivalent to 3 STONES and 1 KEITHS
and 9 MICKS and 6 RONS.
ted@brownsugar Programs 7:54pm >
```

**Hint:** Make use of `while` loops.

**Another Hint:** Look at the picture below...

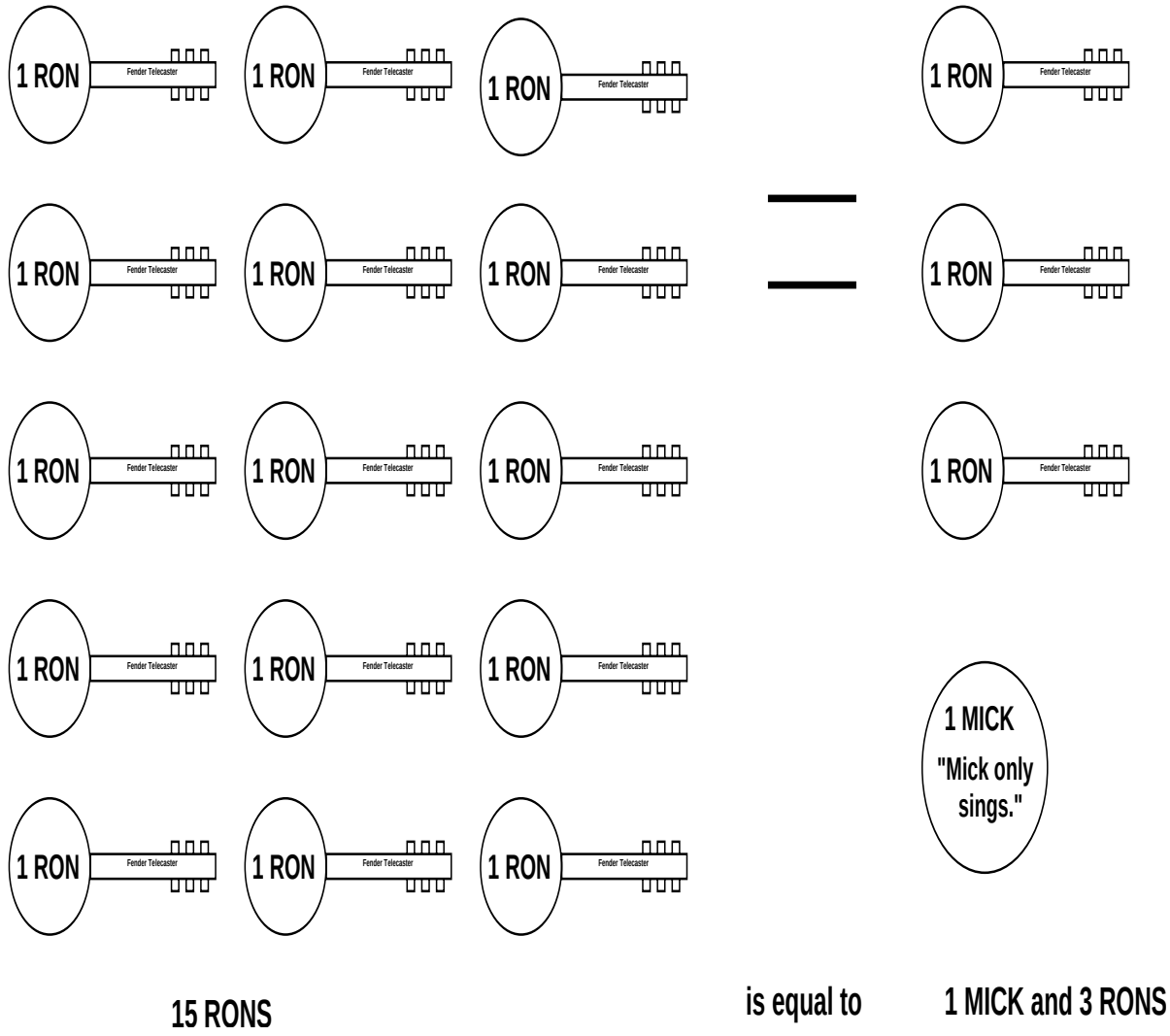


Figure 2: 15 RON coins is equal to 1 MICK coin and 3 RON coins.

**Yet another hint:** 42 MICK coins is the same as 2 KEITH coins and 2 MICK coins.

To help you get started, here is a portion of the program:

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

int main()
{

int stones;
int keiths;
int micks;
int rons;

cout << "Enter the amount of STONES that you have ";
cin >> stones;
cout << "Enter the amount of KEITHS that you have ";
cin >> keiths;
cout << "Enter the amount of MICKS that you have ";
cin >> micks;
cout << "Enter the amount RONS that you have " ;
cin >>  rons;
// the rest is up to you!
```

**Question 2:** [25 points]

(a) Consider the following recursive function and a main program which invokes the function:

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

void print(int n)
{
  if ( n <= 1 )
  {
    cout << "*" << endl;
  } // if
  else
  {
     for ( int i = 1 ; i <= n ; i++ )
     {
         cout << "*" ;
     }
     cout << endl;
     print(n-1); // make a recursive call
     print(n-2);
  } // else
} // end of function

int main()
{

int stars ;
cout << "Enter the number of stars: " ;
cin >> stars;
print(stars);
return 0;
}
```

Answer the following questions:

(i) What output will be produced by the program if the user enters 3 for the value of variable `stars`?

(ii) Draw the function call graph starting with the function `print(3).` Use the notation given in Figure 3 when drawing your call graph.

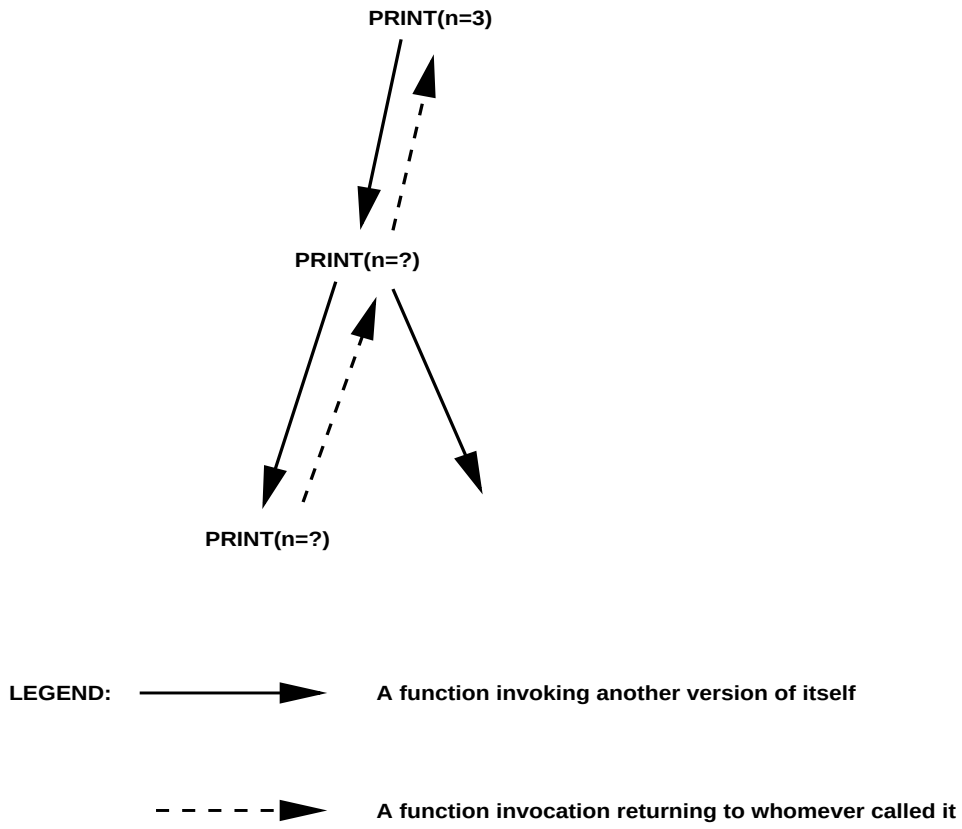


Figure 3: Sample function call graph illustrating notational convention to be used.

(iii) How many times will function `print` be called with argument `n = 1` ? Hint: use your function call graph to help you in answering this question.

(b) Repeat parts (i), (ii), and (iii) if the user enters 4 for the value of variable `stars`.

**Question 3:** [25 points]

Answer TRUE or FALSE to the following questions:

(a) Arrays are always passed by value to a function. TRUE or FALSE?

(b) It is legal (legal in the sense that no compile-time error will be reported) to compare two variables of a given `struct` type directly for equality. TRUE or FALSE?

(c) A `while` loop is always executed at least one time. TRUE or FALSE?

(d) Round off errors do not occur with the `double` data type, therefore it is safe to compare two variables of type double directly for equality with each other. TRUE or FALSE?

(e) A function may not return a user-defined `struct` type. TRUE or FALSE?

(f) Identify what is incorrect in the following function prototype:

```
int let_it_bleed(int [], int , int , int ); // can also let it be
```

(g) What output will be produced by the following program?

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

int main()
{
 cout << "Mick and Keet" << endl;
 return 0;
}
```

**Question 4:** [25 points]

Write a complete C++ function which receives **2 arguments**:

• a one dimensional integer array,
• an integer which specifies the size of the array,

The function is to return an integer which will contain the sum of all the array elements which have an ***even*** array **index**.

The following program gives an example of how this function would be used:

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

// return type and arguments in the prototype for the function are
// intentionally left unspecified, as this is up to
// you to determine how to specify !

 ____ freddy(_____  , ______ );

int main()
{

//   ---------------------
//   | 1 | 2 | 3 | 4 | 5 |  = the array itself
//   ---------------------
//     0   1   2   3   4    = the array indices

 int keet[5] = {1,2,3,4,5}; // Good old Freddy Sessler used to pronounce
```

```
 int sum_of_even_keets;     // 'Keith' as 'Keet', poor Freddy departed
                            // this world on Keet's birthday a few years
                            // ago.
 sum_of_even_keets = freddy(keet, 5);
 cout << "Even keets add up to: " << sum_of_even_keets << endl;
}
```

The output for the sample array in `main()` would be:

```
Even keets add up to: 9
```

**Bonus Question:** [0 points]

Why can't Maybelline be true?