Generate integer from 1 to 7 with equal probability

Given a function foo() that returns integers from 1 to 5 with equal probability, write a function that returns integers from 1 to 7 with equal probability using foo() only. Minimize the number of calls to foo() method. Also, use of any other library function is not allowed and no floating point arithmetic allowed.

Solution:
We know foo() returns integers from 1 to 5. How we can ensure that integers from 1 to 7 occur with equal probability?
If we somehow generate integers from 1 to a-multiple-of-7 (like 7, 14, 21, …) with equal probability, we can use modulo division by 7 followed by adding 1 to get the numbers from 1 to 7 with equal probability.

We can generate from 1 to 21 with equal probability using the following expression.

```
5*foo() + foo() -5
```

Let us see how above expression can be used.
1. For each value of first foo(), there can be 5 possible combinations for values of second foo(). So, there are total 25 combinations possible.
2. The range of values returned by the above equation is 1 to 25, each integer occurring exactly once.
3. If the value of the equation comes out to be less than 22, return modulo division by 7 followed by adding 1. Else, again call the method recursively. The probability of returning each integer thus becomes 1/7.

The below program shows that the expression returns each integer from 1 to 25 exactly once.

```
#include <stdio.h>

int main()
{
    int first, second;
    for ( first=1; first<=5; ++first )
        for ( second=1; second<=5; ++second )
            printf ("%d \n", 5*first + second - 5);
    return 0;
}
```

Output:

```
1
2
.
.
24
25
```

The below program depicts how we can use foo() to return 1 to 7 with equal probability.

```
#include <stdio.h>

int foo() // given method that returns 1 to 5 with equal
{
    // some code here
}

int my_rand() // returns 1 to 7 with equal probability
{
    int i;
    i = 5*foo() + foo() - 5;
    if (i < 22)
        return i%7 + 1;
    return my_rand();
}

int main()
{
    printf ("%d ", my_rand());
    return 0;
}
```