0522042 Programlama Dilleri II
Lab Session Week 2
03/03/2011

Y.T.Ü. Matematik Müh.
### Last week’s Solution

```c
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

int iteration(int);

int main()
{
    int n;
    for (n = 2; n < 100; n++)
        printf("%d takes %d \ steps to return to 1\n", n, iteration(n));
    getch();
    return 0;
}

int iteration(int currn)
{
    int steps = 0;
    while (currn != 1)
    {
        if (2*(currn/2) == currn)
            currn /= 2;
        else
            currn = currn*3 + 1;
        steps++;
    }
    return steps;
}
```

Write a program that performs the following steps for a starting positive integer n which is greater than one:

1) If the value of n is even then n becomes n/2 otherwise n becomes n*3 + 1
2) If the new value of n is greater than one then return to step 1

Hint You can detect if a number is even by using the modulo operator %. (a1 % a2) gives the remainder when a1 is divided by a2.

Count how many times this iteration is performed before the value of n returns to one. For example, if we start with 5 then this becomes 16 then 8, 4, 2 and finally 1. Five iterations in all.

i) How many iterations does it take 7, 8 and 9 to return to 1.

ii) Can you find any values of n which are relatively small (< 30) but take a long time to get back to one?
Fibonacci Numbers & Factorial

```c
int fib(int num)
/* Fibonacci value of a number */
{
    switch(num) {
    case 0:
        return(0);
        break;
    case 1:
        return(1);
        break;
    default: /* Including recursive calls */
        return(fib(num - 1) + fib(num - 2));
        break;
    }
}
```

unsigned int factorial(unsigned int n)
{
    if (n <= 1)
        return 1;
    else
        return n * factorial(n-1);
}

<table>
<thead>
<tr>
<th>Input Value</th>
<th>Number of times fib is called</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>6</td>
<td>25</td>
</tr>
<tr>
<td>7</td>
<td>41</td>
</tr>
<tr>
<td>8</td>
<td>67</td>
</tr>
<tr>
<td>9</td>
<td>109</td>
</tr>
<tr>
<td>10</td>
<td>177</td>
</tr>
</tbody>
</table>
The goal of today’s session:
➢ To practice and get familiar with writing recursive functions in C programming language

Today’s Example problem to work on:

Write a recursive function that performs the following steps for a starting positive integer n which is greater than one:

1) If n is even, let n' = n/2
2) If n is odd, let n' = 3n + 1

Counting the steps, we start:

1) If n is even, let n' = n/2
2) If n is odd, let n' = 3n + 1

Örneğin, dışarıdan girilen n değeri 5 için; sırasıyla 16, 8, 4, 2 ve 1 olmak üzere toplam 5 iterasyon yapılmıştır. n' nin 5

Soru 3) Kendine gönderilen bir pozitif n tamsayısi için aşağıdaki tanımlanan işlemleri yaparak, bu işlemlerin kaç defa tekrar edildiğini sonuç olarak geri döndürecek bir (öz)yinelemeli (recursive) fonksiyon yazınız. (35 puan)
Homeworks

1. What is the difference between recursion and iteration?

2. Write and test both recursive and iterative C functions $a^b$ (without using pow function) for given integers $a$ and $b$

3. Write and test both recursive and iterative functions that converts a base-10 given integer $a$ to its base-2 (binary) equivalent

4. Implement a recursive function to sum first $n$ term of the following serie for given $x$. Compare the obtained result with $\sin(x)$ function in the library math.h

$$\sum_{m=0}^{\infty} \frac{x^m}{m!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots =$$