Introducción a la Programación y Computación 1 Sección E

Ing. MSc. Neftalí Calderón

```
int encriptar(int n) {
   int numero = n;
   int digitos[4];
   int temp;

   for (int i=3; i>=0; i--) {
      digitos[i] = numero%10;
      numero = numero/10;
   }
   for (int i=0; i<4; i++) {
     digitos[i] = (digitos[i]+1)%10;
   }
   temp = digitos[0];
   digitos[0] = digitos[2];
   digitos[2] = temp;

   temp = digitos[1];
   digitos[1] = digitos[3];
   digitos[3] = temp;

   numero = digitos[3] + digitos[2]*10 + digitos[1]*100 + digitos[0]*1000;
   return(numero);
}
```

7381
8492
9284

7391%10 = 1 7391/10 = 739
739%10 = 9 739/10 = 73
73%10 = 3 73/10 = 7
7 % 10 = 7

```
int factorial(int n) {
   int resultado = 1;

   if (n == 0) {
      return(1);
   }
   else {
      for(int i=1; i<=n; i++) {
          resultado = resultado * i;
      }
   }
   return(resultado);
}


main () {
   leer(n);
   imprimir(factorial(n));
}
```

```
int factorial(int n) {
   if (n == 0) {
      return(1);
   }
   else {
      return(n * factorial(n-1));
   }
}
```

!2 = 2*1
!3 = 3*2*1
!4 = 4*3*2*1
!1 = 1

!4 = 4*!3
!3 = 3*!2

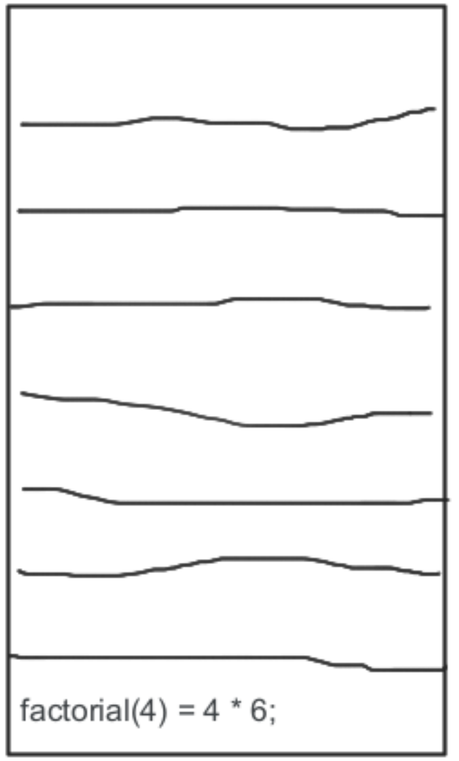
factorial(4) = 4 * factorial(3);
factorial(3) = 3 * factorial(2);
factorial(2) = 2 * factorial(1);
factorial(1) = 1 * factorial(0);
factorial(0) = 1

factorial(4) = 4 * 6;
factorial(3) = 3 * 2;
factorial(2) = 2 * 1;
factorial(1) = 1 * 1;
factorial(0) = 1

factorial(3) = 3 *2;
factorial(2) = 2 * 1

factorial(1) = 1 * 1;

factorial(0) = 1



```
int multiplicar(int a, b) {
   int resultado = 0;

   for (int i=0; i<b; i++) {
      resultado = resultado + a;
   }

   return(resultado);
}
```

```
int multiplicar(int a, b) {
   if (b == 1) {
      return(a);
   }
   else {
      return(a + multiplicar(a, b-1));
   }
}
```

3, 4

multiplicar(3, 4) = 3 + multiplicar(3, 3);
multiplicar(3, 3) = 3 + multiplicar(3, 2);
multiplicar(3, 2) = 3 + multiplicar(3, 1);
multiplicar(3, 1) = 3

multiplicar(3, 4) = 3 + 9
multiplicar(3, 3) = 3 + 6
multiplicar(3, 2) = 3 + 3
multiplicar(3, 1) = 3

1, 1, 2, 3, 5, 8, 13....
1 2 3 4 5 6 7....

sig = ant1 + ant2

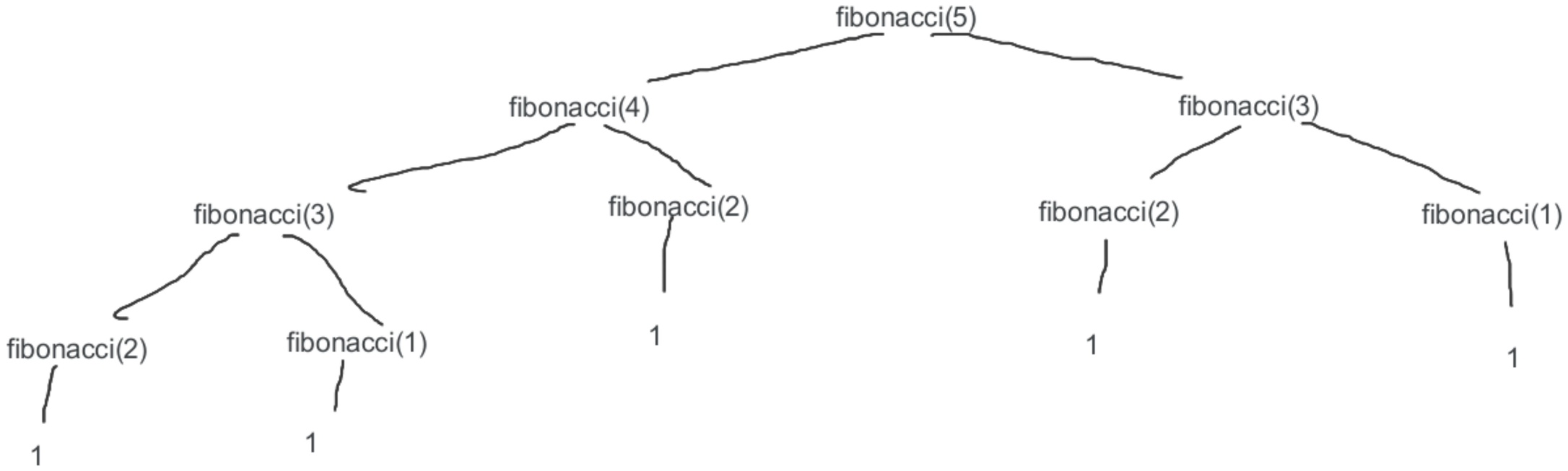
```
int fibonacci(int n) {
   int ant1 = 1;
   int ant2 = 0;
   int sig = 1;

   for (int i=1; i<n; i++) {
      sig = ant1 + ant2;
      ant2 = ant1;
      ant1 = sig;
   }
   return(sig);
}
```

```
int fibonacci(int n) {
   if (n <= 2) {
      return(1);
   }
   else {
      return(fibonacci(n-1) + fibonacci(n-2));
   }
}
```

fibonacci(5) = fibonacci(4) + fibonacci(3);
fibonacci(4) = fibonacci(3) + fibonacci(2);
fibonacci(3) = fibonacci(2) + fibonacci(1);
fibonacci(2) = 1
fibonacci(1) = 1

fibonacci(5) = 3 + 2
fibonacci(4) = 2 + 1
fibonacci(3) = 1 + 1
fibonacci(2) = 1
fibonacci(1) = 1


fibonacci(5)
fibonacci(4)
fibonacci(3)
fibonacci(3)
fibonacci(2)
fibonacci(2)
fibonacci(1)
fibonacci(2)
fibonacci(1)
1
1
1
1
1