

# Chapter 6

# Recursion

مذكرات شرح وتمارين محلولة، امتحانات سابقة للعديد من  
المواد أدناه متاحة مجاناً على الموقعين المذكورين

لن تكون خاسراً أبداً إلا عندما  
تتوقف عن محاولة تحقيق الفوز.

لاستلام نسخ إلكترونية من نوات الموقع مجاناً على إيميلك قم بزيارة [eng-hs.net](http://eng-hs.net)

النوتات متوفرة مجاناً في كل من تصوير الفرع أمام الهندسة أسفل صالون رنيم 24814916 أو تصوير الجمعية الرئيسية بالسرداب أسفل بيانو 24926388

النوتات متوفرة مجاناً بالموقعين [eng-hs.com](http://eng-hs.com), [eng-hs.net](http://eng-hs.net)

(1)

م. حمادة شعبان 9 4444 260 [info@eng-hs.com](mailto:info@eng-hs.com)

1. Consider the following algorithm:

```

algorithm fun1 (x <integer>)
1  if (x < 5)
    1  return (3 * x)
2  else
    1  return (2 * fun1 (x - 5) + 7)
3  end if
end fun1
    
```

What would be returned if fun1 is called as

- fun1 (4)?
- fun1 (10)?
- fun1 (12)?

a)  $3 * 4 = 12$

b)  $2 * \text{fun1}(5) + 7 = 2 * 7 + 7 = 21$

↑

$2 * 2 * \text{fun1}(0) + 7 = 2 * 0 + 7 = 7$

↑

$(3 * 0) = 0$

c)  $2 * \text{fun1}(7) + 7 = 2 * 19 + 7 = 45$

↑

$2 * \text{fun1}(2) + 7 = 2 * 6 + 7 = 19$

↑

$3 * 2 = 6$

إذا قدرت على عدوك فاجعل  
العفو عنه شكراً للقدره عليه

2. Consider the following algorithm:

```

algorithm fun2 (x <integer>
                y <integer>)
1  if (x < y)
   1  return -3
2  else
   1  return (fun2 (x - y, y + 3) + y)
3  end if
end fun2
    
```

What would be returned if fun2 is called as

- a. fun2 (2, 7)?
- b. fun2 (5, 3)?
- c. fun2 (15, 3)?

a) -3

$$\begin{array}{l}
 \text{b) fun2 ( 2, 6 ) + 3} \\
 \uparrow \\
 -3
 \end{array}
 = -3 + 3 = 0$$

$$\begin{array}{l}
 \text{c) fun2 ( 12, 6 ) + 3} \\
 \uparrow \\
 \text{fun2 ( 6, 9 ) + 6} \\
 \uparrow \\
 -3
 \end{array}
 = 3 + 3 = 6$$

$$\begin{array}{l}
 \text{fun2 ( 6, 9 ) + 6} \\
 \uparrow \\
 -3
 \end{array}
 = -3 + 6 = 3$$

بنر الصداقة يزداد عمقا كلما أخذنا منه

لاستلام نسخ إلكترونية من نوات الموقع مجاناً على إيميلك قم بزيارة [eng-hs.net](http://eng-hs.net)

النوتات متوفرة مجاناً في كل من تصوير الفرع أمام الهندسة أسفل صالون رنيم 24814916 أو تصوير الجمعية الرئيسية بالسرداب أسفل بيانو 24926388

النوتات متوفرة مجاناً بالموقعين [eng-hs.com](http://eng-hs.com), [eng-hs.net](http://eng-hs.net)

(3)

م. حمادة شعبان 9 4444 260 [info@eng-hs.com](mailto:info@eng-hs.com)



3. Consider the following algorithm:

```

algorithm fun3 (x <integer>
                y <integer>)
1  if (x > y)
    1  return -1
2  elseif (x equal y)
    1  return 1
3  else
    1  return (x * fun3 (x + 1, y))
4  end if
end fun3
    
```

a) -1

b) -1

c)  $4 * \text{fun3}(5, 7)$

$$\begin{aligned}
 & \downarrow \\
 &= 4 * (5 * \text{fun3}(6, 7)) \\
 & \downarrow \\
 &= 4 * 5 * (6 * \text{fun3}(7, 7)) \\
 & \downarrow \\
 &= 4 * 5 * 6 * (1) = 120
 \end{aligned}$$

النجاح الحقيقي هو أن  
تعثر على حياتك المهنية  
في العمل الذي تحبه

لاستلام نسخ إلكترونية من نوات الموقع مجاناً على إيميلك قم بزيارة [eng-hs.net](http://eng-hs.net)

النوات متوفرة مجاناً في كل من تصوير الفرع أمام الهندسة أسفل صالون رنيم 24814916 أو تصوير الجمعية الرئيسية بالسرداب أسفل بيانو 24926388

النوات متوفرة مجاناً بالموقعين [eng-hs.com](http://eng-hs.com), [eng-hs.net](http://eng-hs.net)

(4)

م. حمادة شعبان 9 4444 260 [info@eng-hs.com](mailto:info@eng-hs.com)

8. Write a recursive algorithm that calculates and returns the length of a linked list.

**public:**

```
int List::Length ( )  
{ return Length ( head ); }
```

**private:**

```
int List::Length ( Node *temp )  
{  
    if ( ! temp )  
        return 0;  
    return 1 + Length ( temp → next );  
}
```

طالما لابد من حضورك المحاضرات فلماذا  
لا تخرج منها بأقصى استفادة ممكنة

لاستلام نسخ إلكترونية من نوات الموقع مجاناً على إيميلك قم بزيارة [eng-hs.net](http://eng-hs.net)

النوتات متوفرة مجاناً في كل من تصوير الفرع أمام الهندسة أسفل صالون رنيم 24814916 أو تصوير الجمعية الرئيسية بالسرداب أسفل بيانو 24926388

النوتات متوفرة مجاناً بالموقعين [eng-hs.com](http://eng-hs.com), [eng-hs.net](http://eng-hs.net)

(5)

م. حمادة شعبان 9 4444 260 [info@eng-hs.com](mailto:info@eng-hs.com)

9. Write a recursive algorithm that converts a string of numerals to an integer. For example, "43567" will be converted to 43567.

```
# include <iostream>
int convert ( char *str, int num )
{
    if ( *str == '\0' )

        return num;

    return convert ( str + 1, num * 10 ( *str - 48 ) )
}

int main ( )
{
    char str [ ] = "43567";

    int num;

    num = convert ( str, 0 );

    cout << "Converted Number is: " << num << endl;

    return 0;

}
```

قال حكيم: إذا تعبت في البر فإن  
التعب يزول والبر يبقى، وإن تلذذت  
بالإثم فإن اللذة تزول ويبقى الإثم.

**10.** Write a recursive algorithm to add the first n elements of the series

$$1 + 1/2 + 1/3 + 1/4 + 1/5 + \dots + 1/n$$

```
# include <iostream>
double fun ( int n )
{
    if ( n == 0 )

        return 0;

    return 1.0 / n + fun ( n - 1 );
}

int main ( )
{
    int n;

    double sum;

    cout << "enter an integer: ";

    cin >> n;

    sum = fun ( n );

    cout << "Sum = " << sum << endl;

    return 0;
}
```

الشخص الذي ينتظر فرصة سابعة  
لتحقيق نجاحه بعد أن ضيع ست فرص  
غالباً سيطول انتظاره

**12.** Write a recursive algorithm to check whether a specified character is in a string.

```

bool Search ( char *str, char key )
{
    if ( *str == '\0' )

        return false;

    if ( *str == key )

        return true;

    return Search ( str + 1, key );
}

int main ( )
{
    char str = "Data is one of my best";

    char key = 'e';

    if ( Search ( str, key ) == true )

        cout << key << " is found\n";

    else

        cout << key << " is not found\n";

    return 0;
}
    
```

أموت محبوبا خير لي من  
أن أعيش مكروها



**18.** Write the recursive version of the binary search algorithm.

```

# include <iostream>
int binSearch ( int x [ ], int start, int end, int key )
{
    if ( start > end )
        return - 1;

    int mid = ( start + end ) / 2;

    if ( x [mid] == key )
        return mid;

    if ( key < x [mid] )
        return binSearch ( x, start, mid - 1, key );

    return binSearch ( x, mid + 1, end, key );
}
    
```

أحيانا يكون كل المطلوب هو تغيير  
وجهة النظر لتحويل عمل شاق  
إلى فرصة مثيرة وممتعة

**24.** Write the C++ function for the recursive algorithm that prints the elements of a linked list in reverse order. (See Algorithm 6-4, "Print linked list reverse," on page 281.)

**public:**

```
void List::printReverse ( )
```

```
{ printReverse ( head ); }
```

---

**private:**

```
void List::printReverse ( Node *temp )
```

```
{
```

```
    if ( ! temp )
```

```
        return;
```

```
        printReverse ( temp → next );
```

```
        cout << temp → next << ' ';
```

```
}
```

**يمكنك أن تخذع الآخرين أحياناً،  
لكن ماذا ستقول لنفسك؟**