**HL Unit 5** **– Abstract Data Structures**  
Recursion Quiz 1

|  |  |  |  |
| --- | --- | --- | --- |
| **Question 1** | | | |
| Objectives: | 5.1.1 | Exam Reference: | May-17 10 |

Outline the reason why recursive solutions can be memory intensive. [2]

*Award up to* ***[2 max]****.*

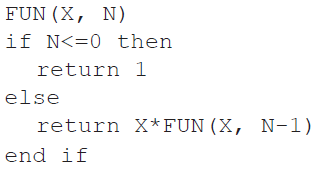
A recursive call involves the use of stacks;

For storing/pushing on/popping out data/ return addresses/return values *etc*;

If many recursive calls are made, the memory usage can be very large;

|  |  |  |  |
| --- | --- | --- | --- |
| **Question 2** | | | |
| Objectives: | 5.1.1, 5.1.2 | Exam Reference: | Nov-16 6 |

Consider the following recursive algorithm FUN(X, N), where X and N are two integers.

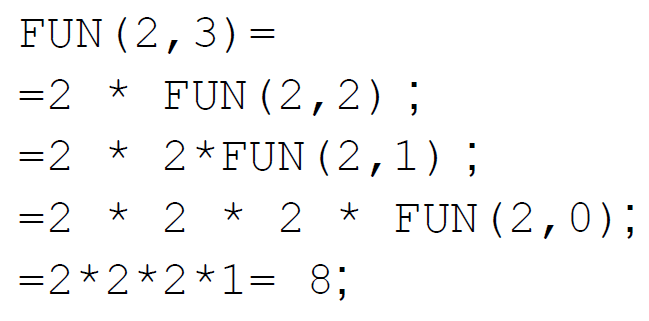


The return statement gives the value that the algorithm generates.

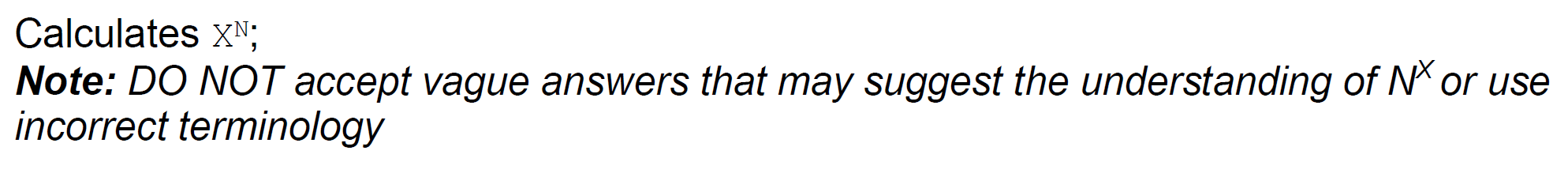
1. Determine how many times multiplication is performed when this algorithm  
   is executed. [1]

N;

1. Determine the value of FUN(2,3), showing all of your working. [3]



1. State the purpose of this recursive algorithm. [1]



|  |  |  |  |
| --- | --- | --- | --- |
| **Question 3** | | | |
| Objectives: | 5.1.1, 5.1.2, 5.1.3 | Exam Reference: | May-16 15 |

1. The letters F0, F1, F2, …, FN , …, where N ≥ 0, are used to identify the N th term of the

sequence of Fibonacci numbers that starts as follows.

0 , 1 , 1 , 2 , 3 , 5 , 8 , 13 , …

With the exception of the leading 0 and 1 (the zeroth term and 1st term), the terms in

the sequence are the sum of the two preceding terms. For example, F5 is the 5th term

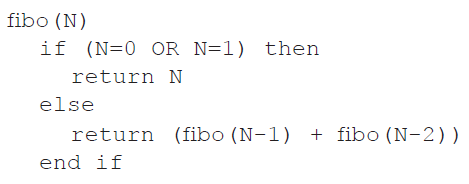
of the sequence, which is 5, and is the sum of the 3rd and 4th terms, which are 2 and 3

respectively.

1. State the value of the 8th term in the sequence. [1]

21

The following method, fibo(N), generates the N th term in the sequence. The return  
 statement returns the value that the method generates.



1. Trace fibo(4), showing the different levels of recursion. [3]

*Award marks as follows up to* ***[3 max]****.*

*Award* ***[1]*** *for evidence of* ***two recursive calls*** *of fibo(2) or of two recursive calls of fibo(0).*

*Award* ***[1]*** *for evidence of* ***three recursive calls*** *of fibo(1).*

*Award* ***[1]*** *for correctly returning values 0 and 1 for fibo(0) and fibo(1) when needed.*

*For example:*

fibo(4)

= (fibo(3)+fibo(2))

= (fibo(2)+fibo(1)) + (fibo(1)+fibo(0))

= ((fibo(1)+fibo(0)) +1) + (1+0)

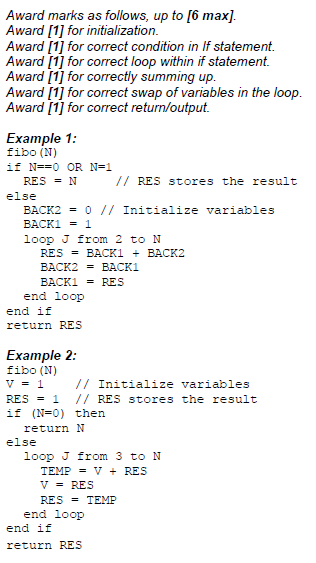
= ((1+0)+1) + (1+0)

= 3

***Note:*** *No marks shall be given if the answer 3 is provided without any tracing, nor if the intermediate values instead of calls to fibo() are given; this is a tracing question.*

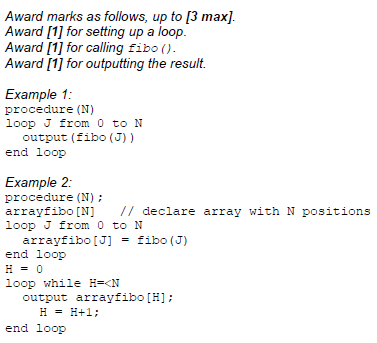
*The order of evaluation of intermediate steps may be different from the one presented in the solution here.*

1. Construct a non-recursive algorithm to generate Fibonacci numbers. [6]



1. Construct an algorithm that will **output** the first N terms of the sequence. You should

use fibo(), the method defined above. [3]



Recursive programs written in high level languages require the use of particular structures to  
 support their execution.

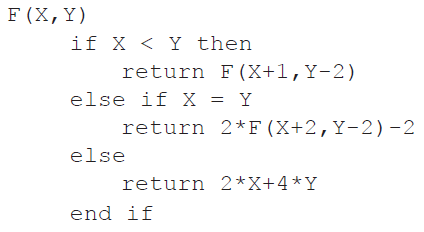
1. Describe how a stack is usually employed in the running of a recursive algorithm. [2]

The current environment (*eg* values/local variables/current address/registers) PUSHED onto the stack when a new recursive call is met;

To be POPPED OFF the stack when the recursive subprogram is completed.

|  |  |  |  |
| --- | --- | --- | --- |
| **Question 4** | | | |
| Objectives: | 5.1.1, 5.1.2 | Exam Reference: | Nov 15 10.f |

Consider the following **recursive** algorithm, in which X and Y are parameters in the  
 method F. The return statement gives the value that the method generates.



1. Determine the value of F(5,11). [5]

