|  |  |
| --- | --- |
| **Data Structures & Algorithms**  Diploma in IT / ISF  Year 2 (2017/18) Semester 4 | **Week 6** |
| **2 Hours** |
| **Tutorial 6 – Recursion** | |

1. Write a recursive function that will compute the sum of the first n integers in an array of at least n integers. Hint: begin with the nth integer.

// the short way

int sum (int array [ ] , int n}

{

If (n==0) //base case

Return 0

Else

return **array[n-1]** + **sum (array, n-1);**

}

2. Describe the problem with the following recursive function:

void printNum(int n)

{

cout << n <<endl;

printNum(n-1);

}

// there is no base case to tell the program

// to stop recurring when the problem is very small.

3. Given an integer n > 0, write a recursive function that returns the sum of 1 through n.

4x - 6

// return the sum of 1 through n

// n < 0 (base case)

// n is a real number >= 0

int computeSum(int sum, int n) {

if (n == 1)

return sum + 1;

else {

computeSum((sum + n), (n - 1));

}

}

4a. Write a recursive C++ function writeLine that writes a character repeatedly to form a line of n characters. For example, writeLine(‘\*’, 5) produces the line \*\*\*\*\*

char printLine(char n, int num){

if (num == 1)

cout << n<< endl;

else{

printLine(n, num -1);

}

b. Now write a recursive function writeBlock that uses writeLine to write m lines of n characters each. For example, writeBlock(‘\*’, 3, 5) produces the output

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

5. Consider the following program:

int f(int n);

int main()

{

cout<< “The value of f(8) is “ << f(8) << endl;

return 0;

}//end of main

/\*\* @pre n >= 0 \*/

int f(int n)

{

cout << “Function entered with n = “ << n<< endl;

switch(n)

{

case 0: case 1: case 2:

return n + 1;

default:

return f(n-2) \* f(n-4);

}//end of switch

}//end of f

Show the exact output of the program. What argument values, if any, could you pass to the function f to cause an infinite recursion?

6. Trace the call f(16) to the following method by showing a stack of activation records:

int f(int n)

{

int result = 0;

if (f <= 2)

result = 1;

else

result = f(n / 2) + f(n / 4);

return result;

} // end f