1. What is a direct recursion?

A recursive function is one that calls itself repetitively until a final call is made that no longer requires a self-call

1. For the remaining questions, use the following recursive function:



|  |
| --- |
| int Smiley (int n){  if (n == 0)  return 3;  else  return n + Smiley(n-1);  } |

* + - What final value is returned from the function call Smiley(5)?



5+4+3+2+1+3

=18

Smiley(5)= 5+13

Smiley(4)= 4+9

Smiley(3)= 3+6

Smiley(2)= 2+4

Smiley(1)= 1+3

Smiley(0)= 3

Smiley(5)= 5+13

Smiley(4)= 4+9

Smiley(3)= 3+6

Smiley(2)= 2+4

Smiley(1)= 1+3

Smiley(0)= 3

* + - What final value is returned from the function call Smiley(0)?

3

* + - What final value is returned from the function call Smiley(-1)?

Infinite loop

* + - Suppose + is changed to \* in the recursive step. What is the answer for Smiley(5)?

1. Is the following function circular? Explain.

int Thor(int pq)

{ if (pq==1) return 0;

else if (pq % 2 != 0) return Thor(pq/2);



else return 1 + Thor(3\*pq + 1);}

4. Consider the following program, (*If there is an error in the code and you need to correct it accurately*)

#include <stdio.h>

int power(int n1);

int main() {

int a, result;

printf("Enter base number: ");

scanf("%d", &base);

printf("Enter power number(positive integer): ")

scanf("%d", &a);

result = power(base, a);

printf("%d^%d = %d", base, a, result);

return 0;

}

int power(int base, int a, int result) {

if (a != 0)

return (power \* power(base, a + 1));

else

return 1;

}

5. Write a program in C to print first 50 natural numbers using recursion.

The natural numbers are : 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50.