Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

Examination: Assignment 4 Session: Spring 2021

Year/Semester: 1/1 (A1, C1, C2) Course No: CSE1102

Course Title: Elementary Structured Programming Lab

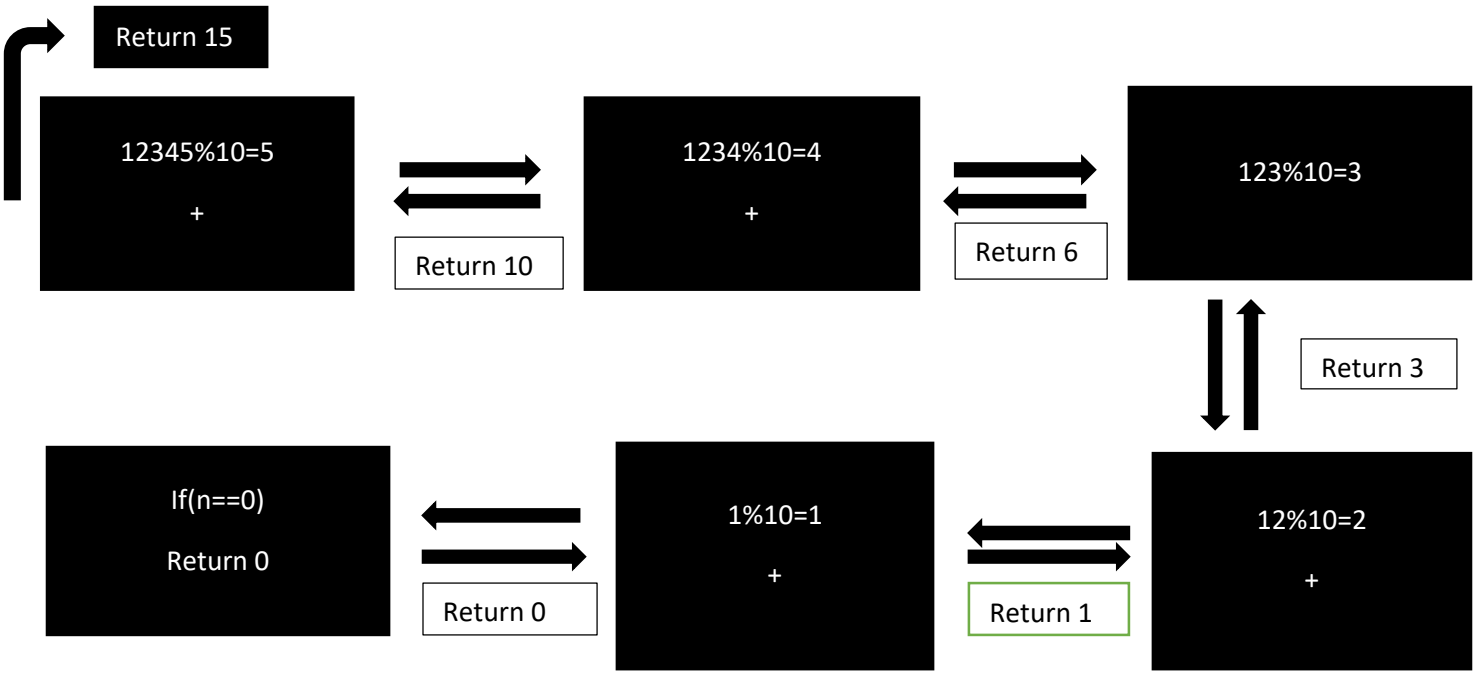
ID_20210104111

1. A 5 digit positive integer is entered through the keyboard, write a function to calculate the sum of digits of the 5 digit number using recursion.

SOLVE:

```c
#include <stdio.h>
int sum_of_digit(int n)
{
    return(n > 0 ? (n % 10 + sum_of_digit(n / 10)) : 0);
}
int main()
{
    int num = 12345;
    printf("%d\n", sum_of_digit(num));
    return 0;
}
```

RECURSION TREE:



2. Convert Decimal number to binary using recursion.

SOLVE:

```c
#include<stdio.h>
int decimal_binary(n)
{
    return(n > 0 ? ((n % 2) + 10 * decimal_binary(n / 2)) : 0);
}
int main()
{
    int num;
    printf("Enter a integer number: ");
    scanf("%d", &num);
    printf("decimal %d = binary %d\n",num,decimal_binary(num));
    return 0;
}
```

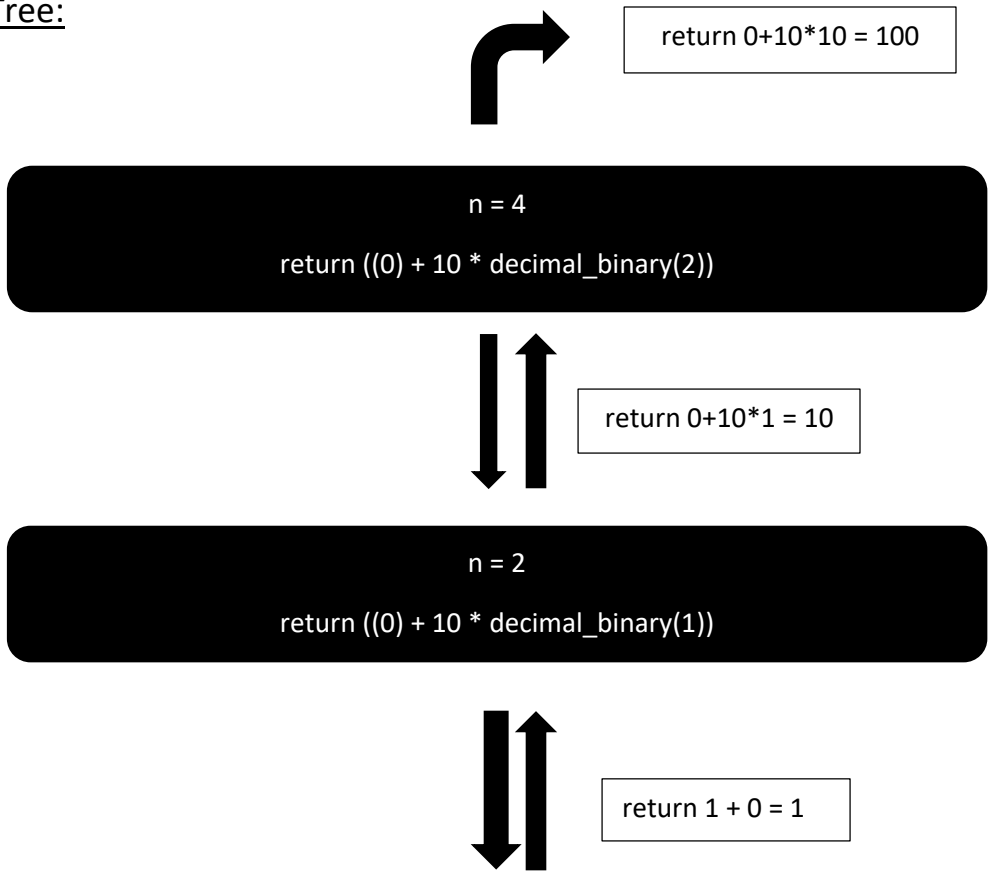
Input:4

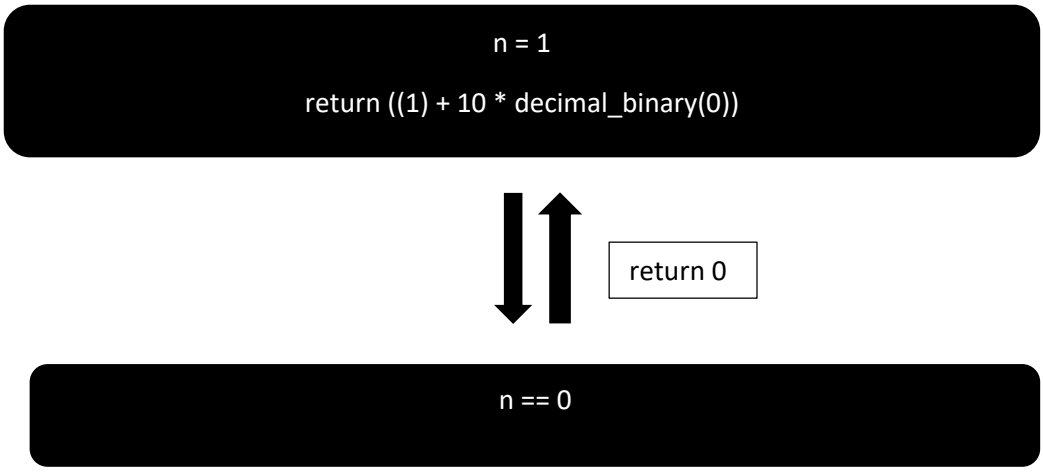
Output:

Enter a integer number: 4

decimal 4 = binary 100

Recursion Tree:





3. Calculate the power of any number using a recursive method. Take the base and exponent values as user input.

SOLVE:

```c
#include <stdio.h>
int power(int base,int exponent)
{
    return(exponent == 0 ? 1 : base * power(base,exponent - 1));
}
int main()
{
    int base,exponent;
    printf("Enter the value of base and exponent: ");
    scanf("%d %d",&base, &exponent);
    printf("power(%d^%d) = %d",base,exponent,power(base,exponent));
    return 0;
}
```

INPUT:

2,3

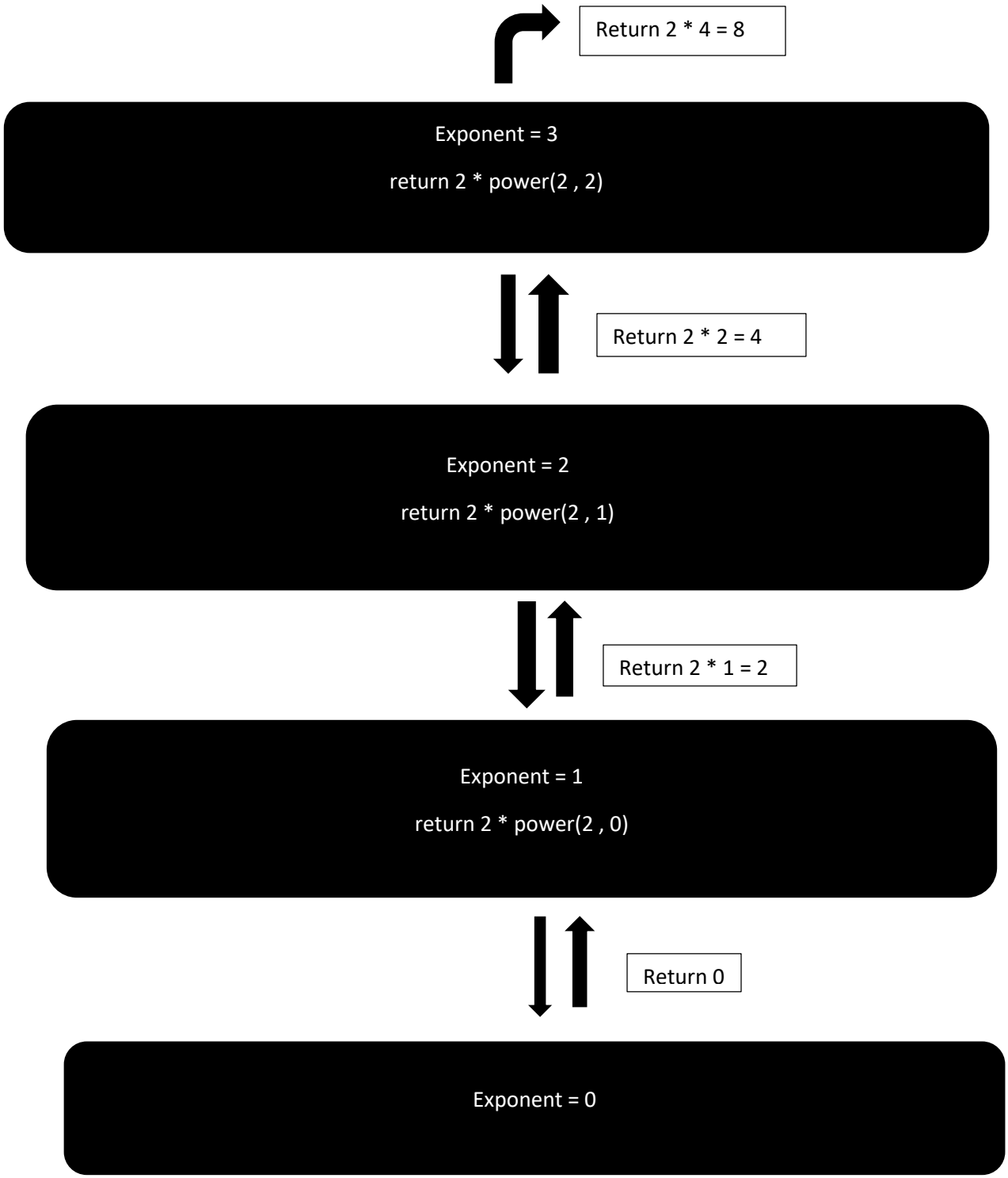
OUTPUT:

Enter the value of base and exponent: 2

3

power(2^3) = 8

RECURSION TREE:


Return 2 * 4 = 8
Exponent = 3
return 2 * power(2 , 2)
Return 2 * 2 = 4
Exponent = 2
return 2 * power(2 , 1)
Return 2 * 1 = 2
Exponent = 1
return 2 * power(2 , 0)
Return 0
Exponent = 0

4. Find the LCM of two numbers using recursion.

SOLVE:

```c
#include<stdio.h>
int lcm(int a,int b)
{
    static int m = 0;
    m = m + b;
    return ( m%a == 0 && m%b == 0) ? m : lcm(a,b);
}
int main()
{
    int num1,num2;
    printf("Enter the first and second number: ");
    scanf("%d %d",&num1, &num2);
    printf("lcm = %d",lcm(num1, num2));
    return 0;
}
```

Input:

6,16

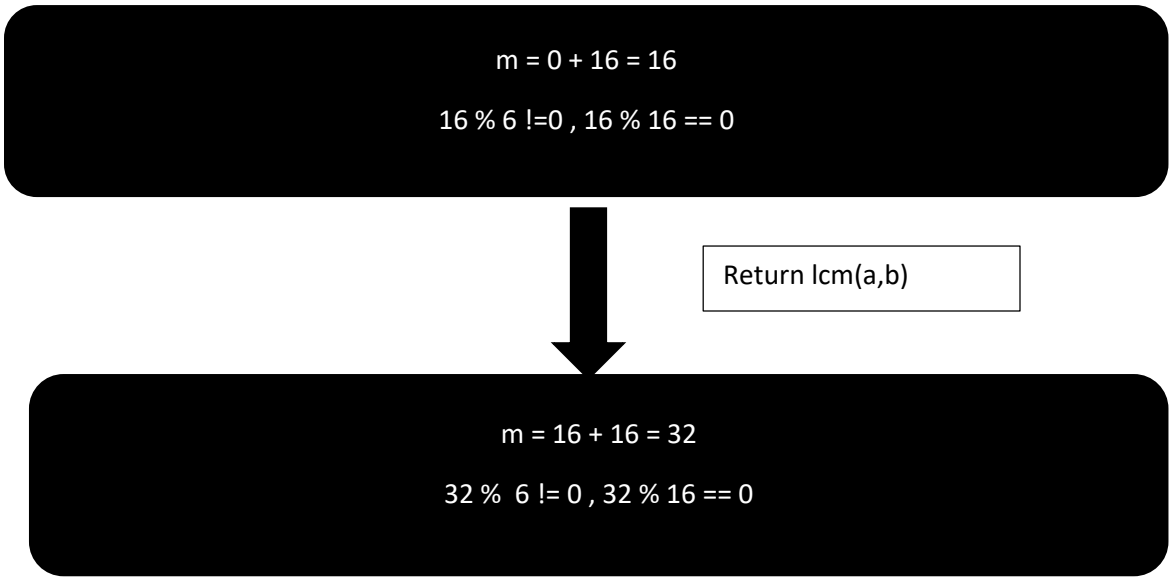
Output:

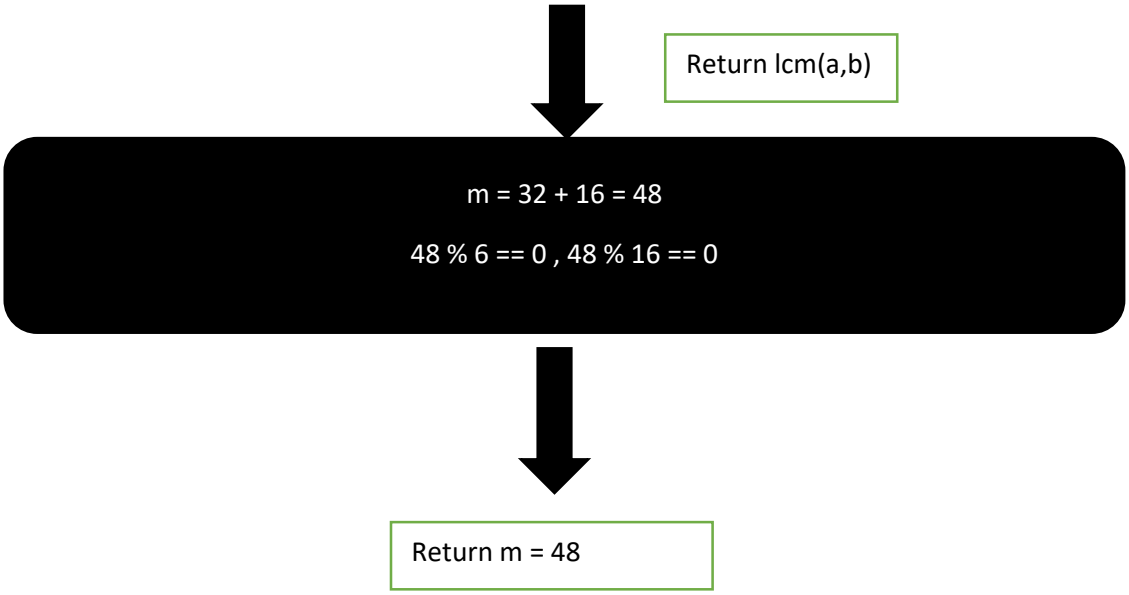
Enter the first and second number: 6

16

lcm = 48

RECURSION TREE:





5.Write a recursive function to obtain the first 25 numbers of a Fibonacci sequence. In a Fibonacci sequence the sum of two successive terms give the third term. Following are the first few terms of the Fibonacci sequence:

1 1 2 3 5 8 13 21 34 55

SOLVE:

```c
#include<stdio.h>
int fibonacci(int n)
{
return (n == 0 || n == 1 ? n : fibonacci(n - 1) + fibonacci(n - 2));
}
void main()
{
    int n, c=0, i;
    printf("Enter number: ");
    scanf("%d", &n);
    printf("Fibonacci Series:\n");
    for(i = 1; i <= n; i++)
    {
        printf("%d\n", fibonacci(c));
        c++;
    }
}
```

INPUT:25

OUTPUT:

Enter number: 25

Fibonacci Series:

0

1

1

2

3

5

8

13

21

34

55

89

144

233

377

610

987

1597

2584

4181

6765

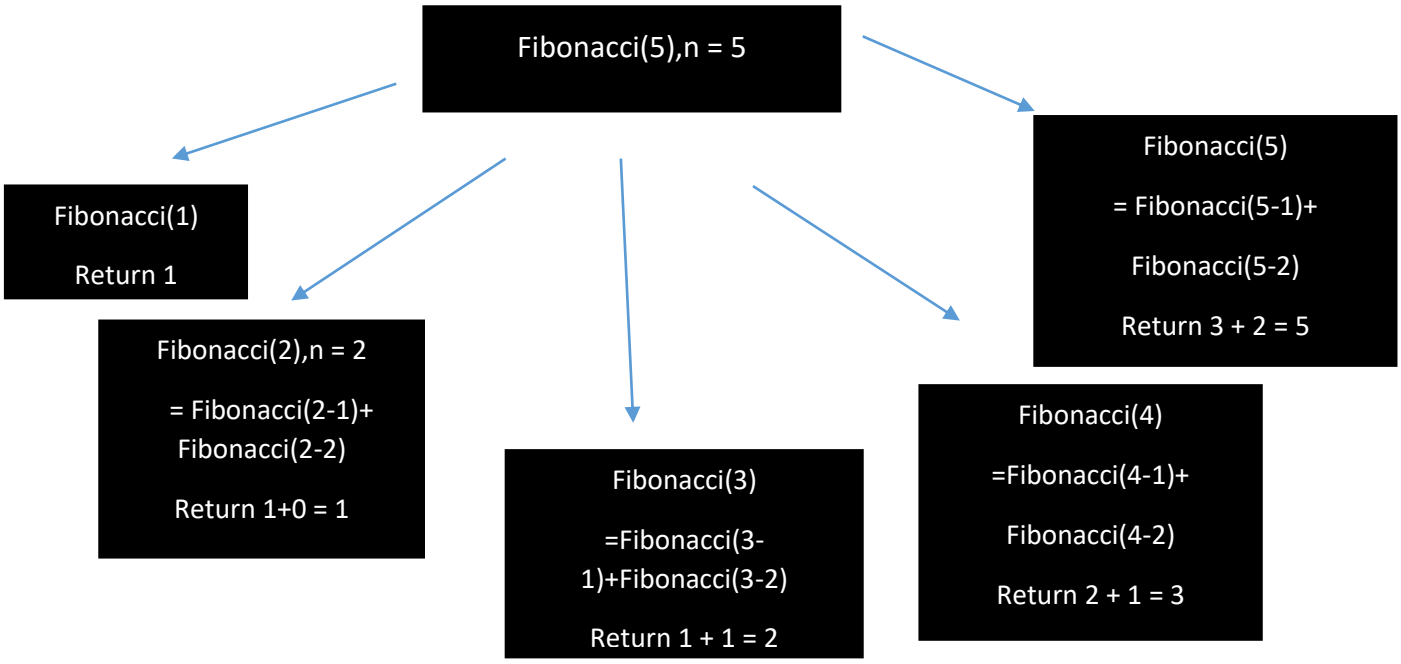
10946

17711

28657

46368

RECURSION TREE:



6. WAP to find out the sum of the following series using recursion, where the values of a, r and n are user input:

a+ar+ar^2+ar^3+……..+ar^(n-1)

SOLVE:

```c
#include <stdio.h>
#include <math.h>

int sum_of_series();

int main()
{
    int a, r, n;
```

```
    scanf("%d %d %d", &a, &r, &n);

    n = n - 1;

    printf("%d", sum_of_series(a, r, n));

    return 0;
}
int sum_of_series(int l, int m, int n)
{
    static int s = 0;

    s += l * pow(m, n);

    n--;
    if (n >= 0)
    {
        sum_of_series(l, m, n);
        return s;
    }
}
```

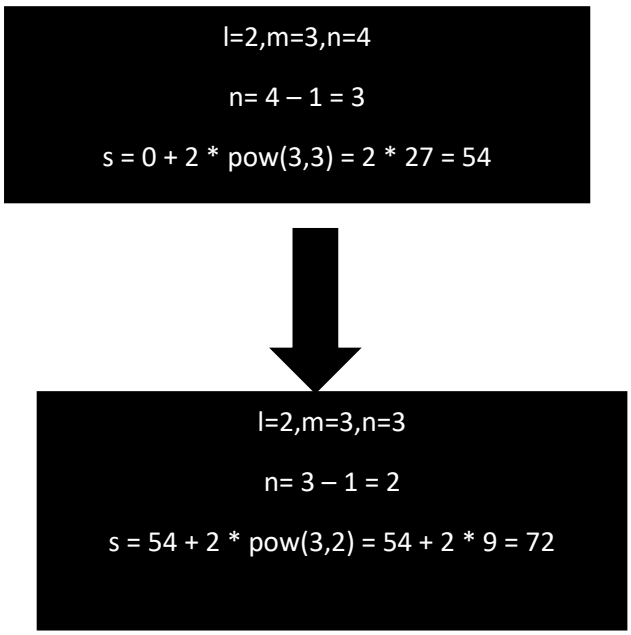
INPUT:

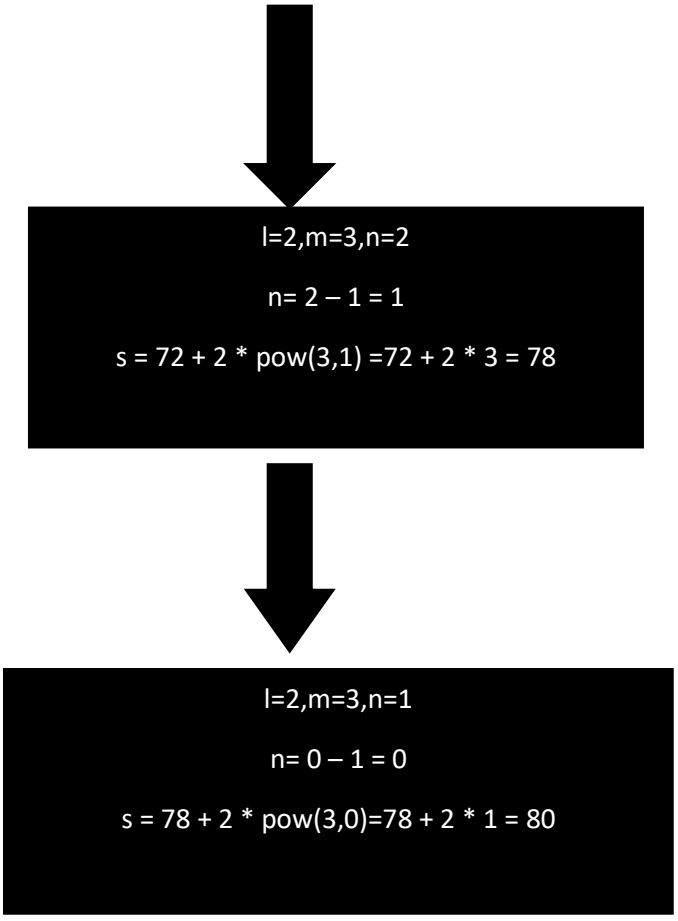
2 3 4

OUTPUT:

80

RECURSION TREE:




l=2,m=3,n=2
n= 2 – 1 = 1
s = 72 + 2 * pow(3,1) =72 + 2 * 3 = 78
l=2,m=3,n=1
n= 0 – 1 = 0
s = 78 + 2 * pow(3,0)=78 + 2 * 1 = 80