Lecture 15 Recursion

1. Factorial
2. #include <iostream>
3. using namespace std;
4. int fact(int n)
5. {
6. // base case
7. if(n==0)
8. {
9. return 1;
10. }
11. // recursive case
12. int smallAns = n \* fact(n-1);
13. return smallAns;
14. }
15. int main() {
16. int n;
17. cin>>n;
18. cout<<fact(n)<<endl;
19. return 0;
20. }

2. Fibonacci series

#include <iostream>

using namespace std;

int fib(int n)

{

    // base case

    if(n==0 || n==1)

    {

        return n;

    }

    // recursive case

    int f1 = fib(n-1); // 4 // 3// 2 // 1

    int f2 = fib(n-2);

    return f1+f2;

}

int main() {

    int n;

    cin>>n;

    cout<<fib(n)<<endl;

    return 0;

}

3. Power

#include <iostream>

using namespace std;

int power(int a, int b)

{

    // base case

    if(b==0)

    {

        return 1;

    }

    // recursive part

    else

    {

        return a\*power(a, b-1);

    }

}

int main() {

    int a,b;

    cin>>a>>b;

    cout<<power(a,b)<<endl;

    return 0;

}

4. Increasing Decreasing order

#include <iostream>

using namespace std;

void decOrder(int n)

{

    if(n==0)

    {

        return;

    }

    cout<<n<<", ";

    decOrder(n-1);

}

void incOrder(int n)

{

    if(n==0)

    {

        return;

    }

    incOrder(n-1);

    cout<<n<<", ";

}

int main() {

    int n = 5;

    decOrder(n);

    incOrder(n);

    return 0;

}