RECURSION

1. PRINT N TO 1

```
#include <iostream>
using namespace std;


void printToN(int n)
{
        if(n == 0)
                return;

        cout<<n<<" ";

        printToN(n - 1);

}

int main() {

        int n = 4;

        printToN(n);

        return 0;
}
```

2. PRINT 1 TO N

```
#include <iostream>
using namespace std;
```

```
void printToN(int n)
{
	if(n == 0)
		return;

	printToN(n - 1);

	cout<<n<<" ";

}

int main() {

	int n = 4;

	printToN(n);

	return 0;
}
```

3. FACTORIAL

```
#include <iostream>
using namespace std;


int fact(int n, int k)
{
	if(n == 0 || n == 1)
		return k;

	return fact(n - 1, k * n);
```

```
}

int main() {

	cout<<fact(3, 1);
}
```

4. NATURAL NUMBER SUM

```
#include <iostream>
using namespace std;


int getSum(int n)
{
	if(n == 0)
		return 0;

	return n + getSum(n - 1);

}

int main() {

	int n = 4;

	cout<<getSum(n);

	return 0;
}
```

5. PALINDROME CHECK

```
#include <iostream>
using namespace std;


bool isPalindrome(string str, int start, int end)
{
        if(start >= end)
                return true;

        return ((str[start]==str[end]) &&
                    isPalindrome(str, start + 1, end - 1));
}

int main() {

        string s = "GeekskeeG";

                printf("%s", isPalindrome(s, 0, s.length() -1) ? "true" :
"false");

        return 0;
}
```

6. SUM OF DIGITS

```
#include <iostream>
using namespace std;


int fun(int n)
{
        if(n < 10)
                return n;
```

```
        return fun(n / 10) + n % 10;
}

int main() {

        cout<<fun(253);

        return 0;
}
```

7. ROPE CUTTING PROBLEM

```
#include <iostream>
using namespace std;


int maxCuts(int n, int a, int b, int c)
{
        if(n == 0)
                return 0;
        if(n <= -1)
                return -1;

        int res = max(maxCuts(n-a, a, b, c),
                 max(maxCuts(n-b, a, b, c),
                 maxCuts(n-c, a, b, c)));

        if(res == -1)
                return -1;

        return res + 1;
}
int main() {
```

```
    int n = 5, a = 2, b = 1, c = 5;

    cout<<maxCuts(n, a, b, c);

    return 0;
}
```

8. **GENERATE SUBSETS**

```
#include <iostream>
using namespace std;


void printSub(string str, string curr, int index)
{
    if(index == str.length())
    {
        cout<<curr<<" ";
        return;
    }

    printSub(str, curr, index + 1);
    printSub(str, curr+str[index], index + 1);
}

int main() {

    string str = "ABC";

   printSub(str, "", 0);

    return 0;
}
```

9. TOWER OF HANOI

```
#include <iostream>
using namespace std;


void ToH(int n, char A, char B, char C)
{
   if (n == 1)
   {
      cout<<"Move 1 from " <<  A << " to " << C << endl;
      return;
   }
   ToH(n-1, A, C, B);
   cout<<"Move " << n << " from " <<  A << " to " << C << endl;
   ToH(n-1, B, A, C);
}

int main() {

        int n = 2;

   ToH(n, 'A', 'B', 'C');

        return 0;
}
```

10. JOSEPHUS PROBLEM

```
#include <iostream>
using namespace std;


int jos(int n, int k)
```

```
{
	if(n == 1)
		return 0;
	else
		return (jos(n - 1, k) + k) % n;
}

int myJos(int n, int k)
{
	return jos(n, k) + 1;
}

int main() {

	cout<<myJos(5, 3);

	return 0;
}
```

11. SUBSET SUM PROBLEM

```
#include <iostream>
#include <limits.h>
using namespace std;

int countSubsets(int arr[], int n, int sum)
{
	if(n==0)
		return sum==0? 1 : 0;

	return countSubsets(arr, n-1, sum) + countSubsets(arr, n-1,
sum - arr[n-1]);
}
```

```
int main() {

        int n = 3, arr[]= {10, 20, 15}, sum = 25;

        cout<<countSubsets(arr, n, sum);

        return 0;
}
```

12. PRINTING ALL PERMUTATIONS OF A STRING