1) Fibbonocci Series

```
#include <iostream>
#include<vector>
#include<cmath>
using namespace std;
class Solution {
public:
 static std::vector<int> generateFibonacci(int n) {
 std::vector<int> result;
 int i,a=1,b=0,temp=0;
 for(i=0;i<n;i++)
 {
    result.push_back(temp);
    b=temp;
    temp+=a;
    a=b;
 }

 return result;
 }
};
```

```
int main() {
 int n = 5;
 std::vector<int> result = Solution::generateFibonacci(n);
 std::cout << "Fibonacci Series: ";
 for (int num : result) {
 std::cout << num << " ";
 }
 std::cout << std::endl;
 return 0;
}
```

2) Armstrong Number

```
#include<iostream>
#include<cmath>
using namespace std;
class Solution {
public:
 static bool isArmstrong(int n) {
 bool result = false;
 int temp=0,count=0,sum=0;
 temp=n;
 while(temp>0)
 {
     count++;
     temp/=10;
 }
 temp=n;
 while(n>0){
     sum+=pow(n%10,count);
     n/=10;
 }
 if(sum==temp) result=true;
 return result;
 }
};
int main() {
 int n = 153;
 bool result = Solution::isArmstrong(n);
 std::cout << "Is Armstrong: " << (result ? "Yes" : "No") <<
std::endl;
 return 0;
}
```

3)Add and Subtract two numbers without using arithemetic

operators

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

class Solution {
public:
 static int add(int a, int b) {
 int result;
 a+=b;
 return result=a;
 }
 static int subtract(int a, int b) {
 int result;
 a-=b;
 return result=a;
 }
};
int main() {
 int a = 15, b = 10;
 int addResult = Solution::add(a, b);
 int subtractResult = Solution::subtract(a, b);
 std::cout << "Addition result: " << addResult << std::endl;

 std::cout << "Subtraction result: " << subtractResult << std::endl;

 return 0;

}
```

4)Binary to Decimal

```
#include<iostream>

#include<cmath>

using namespace std;

class Solution {

public:

 static int binaryToDecimal(const std::string& binary) {

 int result = 0,i,j=0;

 for(i=binary.length()-1;i>=0;i--)

 {

     if(binary[i]=='1') result+=1*pow(2,j++);

     else result+=0*pow(2,j++);

 }

 return result;

 }

};

int main() {
 std::string binary = "1101";
 int result = Solution::binaryToDecimal(binary);
 std::cout << "Decimal: " << result << std::endl;
 return 0;
}
```

5)Decimal to Binary

```
#include<iostream>
#include<string>
using namespace std;
class Solution {
public:
 static std::string decimalToBinary(int decimal) {
 std::string result;
 while(decimal>0){
     result=((decimal%2)? "1" : "0" )+result;
     decimal/=2;
 }
 return result;
 }
};
int main() {
 int decimal = 13;

 std::string result = Solution::decimalToBinary(decimal);

 std::cout << "Binary: " << result << std::endl;

 return 0;

}
```