**National Collegiate Programming Competition**

Millitary Institute of Science and Technology

**CodeBook**

**GDC**

int gcd(int n, int m) {

if (n % m == 0) return m;

return gcd(m, n % m);

}

**LCM**

int lcm(int a, int b) { return (a \* b) / gcd(a, b); }

**Leap Year**

bool isLeapYear(int year) {

if ((year % 400 == 0) || (year % 100 != 0 && year % 4 == 0)) {

return true;

} else {

return false;

}

}

**math.h**

» sqrt(n) // square root

» fabs(n) // absolute

» sin(n), cos(n), tan(n)

» asin(n), acos(n), atan(n) // inverse

» atan2(y, x)

» pow(n, m)

» exp(n)

» log(n), log10(n)

» floor(n), ceil(n)

**prime number**

bool isPrime(int n) {

if (n <= 1) return false;

for (int i = 2; i < n; i++)

if (n % i == 0)

return false;

return true;

}

**BigMod**

int bigMod(int a, int b, int M) {

if (b == 0) return 1 % M;

int x = bigMod(a, b / 2, M);

x = (x \* x) % M;

if (b % 2 == 1) x = (x \* a) % M;

return x;

}

1. Moduler Inverse
2. Factorial
3. Combination

**Fibonacci Number**

int fibonacci(int n) {

if (n == 0) return 0;

if (n == 1) return 1;

return fibonacci(n - 1) + fibonacci(n - 2);

}

1. Probability
2. Expectation
3. Big Integer
4. Cycle
5. Gaussian elimination
6. Insertion sort
7. Selection sort
8. Bubble sort
9. Merge sort
10. Counting sort
11. STL::sort
12. Binary search
13. Ternary search
14. Backtracking
15. Permutation
16. Linked list
17. Stack
18. Queue
19. Graph
20. Tree
21. Binary search tree
22. Heap & Priority queue
23. Minimum Spanning Tree
    1. Prim’s Algorithm
    2. Kruskal’s Algorithm
24. Huffman Coding
25. BFS
26. DFS
27. Hashing