Task 1

#include <iostream>

#include <string>

using namespace std;

void inputFunc(int N, int element) { // O(n)

int el;

if (N > 0) {

cin >> el;

if (element > el) {

element = el;

}

inputFunc(N - 1, element);

}

else {

cout << element;

}

}

int main() { // O(1)

int N;

cin >> N;

int element;

cin >> element;

inputFunc(N - 1, element);

return 0;

}

Task 2

#include <iostream>

#include <string>

using namespace std;

int inputFunc(int N, int element) { // O(n)

int el;

if (N > 0) {

cin >> el;

element += el;

inputFunc(N - 1, element);

}

else {

return element;

}

}

int main() { // O(1)

int N;

float avg;

cin >> N;

int element;

cin >> element;

avg = inputFunc(N - 1, element);

cout << avg / N;

return 0;

}

Task 3

#include <iostream>

#include <string>

using namespace std;

int main() { // O(n)

int number, count = 0;

cin >> number;

for (int i = 2; i < number; i++) {

if (number % i == 0) {

cout << "Composite";

count++;

break;

}

}

if (count == 0) {

cout << "Prime";

}

return 0;

}

Task 4

#include <iostream>

#include <string>

using namespace std;

void findFactorial(int number, int fact) { // O(n!)

fact \*= number;

if (number > 1) {

findFactorial(number - 1, fact);

}

else {

cout << fact;

}

}

int main() { //O(1)

int number, fact = 1;

cin >> number;

findFactorial(number, fact);

return 0;

}

Task 5

#include <iostream>

#include <string>

using namespace std;

void findFib(int num, int Fib1, int Fib2) { //O(2^n)

int betFib;

if (num > 1) {

betFib = Fib2;

Fib1 += Fib2;

Fib2 = Fib1;

Fib1 = betFib;

findFib(num - 1, Fib1, Fib2);

}

else {

cout << Fib1 + Fib2;

}

}

int main() { //O(1)

int number, Fib1 = 0, Fib2 = 1;

cin >> number;

findFib(number - 2, Fib1, Fib2);

return 0;

}

Task 6

#include <iostream>

#include <string>

using namespace std;

int main() { //O(1)

int number, power;

cin >> number >> power;

int timeNum = number;

for (int i = 1; i < power; i++) { //O(n^m)

number \*= timeNum;

}

cout << number;

return 0;

}

Task 7

#include <iostream>

#include <string>

using namespace std;

int main() {

string str;

char b;

int a = 0;

cin >> str;

for (int j = 0; j < str.length(); j++) { // O(n!)

for (int i = 0; i < str.length() - 1; i++) {

b = str[i];

str[i] = str[i + 1];

str[i + 1] = b;

cout << str << endl;

cout << i << endl;

}

}

return 0;

}

Task 8

#include <iostream>

#include <string>

using namespace std;

int main() { // O(1)

string str;

int a = 0;

cin >> str;

for (int i = 0; i < str.length() - 1; i++) {

if (str[i] > '9') {

a++;

cout << "No";

break;

}

}

if (a == 0) {

cout << "Yes";

}

return 0;

}

Task 9

#include <iostream>

#include <string>

using namespace std;

int factorial(int num) { // O(n!)

int count = 1;

for (int i = 1; i <= num; i++) {

count \*= i;

}

return count;

}

int main() { // O(1)

int n, k;

cin >> n >> k;

cout << factorial(n) / (factorial(n - k) \* factorial(k));

return 0;

}

Task 10

#include <iostream>

#include <string>

using namespace std;

int main() {

int a, b;

cin >> a >> b;

int i = 1, c;

while (b != 0) { // O(GCD(a,b))

while (b \* i <= a) {

i++;

}

c = b;

b = a - (b \* (i - 1));

a = c;

i = 1;

}

cout << a;

return 0;

}