{

    // Place your snippets for cpp here. Each snippet is defined under a snippet name and has a prefix, body and

    // description. The prefix is what is used to trigger the snippet and the body will be expanded and inserted. Possible variables are:

    // $1, $2 for tab stops, $0 for the final cursor position, and ${1:label}, ${2:another} for placeholders. Placeholders with the

    // same ids are connected.

    // Example:

    "boilerplate": {

        "prefix": "boilerplatecode",

        "body": [

            "#include <bits/stdc++.h>",

            "#include<sstream> ",

            "using namespace std;",

            "template<typename A, typename B> ostream& operator<<(ostream &os, const pair<A, B> &p) { return os << '(' << p.first << ', ' << p.second << ')'; }",

            "template<typename T\_container, typename T = typename enable\_if<!is\_same<T\_container, string>::value, typename T\_container::value\_type>::type> ostream& operator<<(ostream &os, const T\_container &v) { os << '{'; string sep; for (const T &x : v) os << sep << x, sep = ', '; return os << '}'; }",

            "void dbg\_out() { cerr << endl; }",

            "template<typename Head, typename... Tail> void dbg\_out(Head H, Tail... T) { cerr << ' ' << H; dbg\_out(T...); }",

            "#ifdef LOCAL",

            "#define dbg(...) cerr << '(' << #\_\_VA\_ARGS\_\_ << '):', dbg\_out(\_\_VA\_ARGS\_\_)",

            "#else",

            "#define dbg(...)",

            "#endif",

            "#define ar array",

            "#define ll long long",

            "#define ld long double",

            "#define vi vector<int>",

            "#define vvi vector<vector<int>>",

            "#define vl vector<ll>",

            "#define vvl vector<vector<ll>>",

            "#define sza(x) ((int)x.size())",

            "#define all(a) (a).begin(), (a).end()",

            "#define FOR(i,a,b) for( long long int i = a; i<b;i++)",

            "#define pb push\_back",

            "#define ce(x) cout<<x<<endl",

            "int modmul(int a,int b,int m){ a %= m;b %= m; return (a \* b) % m;}",

            "int modadd(int a,int b,int m){ a %= m;b %= m; return (a + b) % m;}",

            "int modsub(int a,int b,int m){ a %= m;b %= m; return (a - b + m) % m;}",

            "int gcd(int a, int b){ if(b == 0) return a; return gcd(b, a % b);}",

            "int expo(int a,int n,int md){ int res=1; while(n){ if(n&1) {res = modmul(res,a,md);--n;} else {a = modmul(a,a,md);n >>= 1;}} return res;}",

            "int expo(int a,int n){ int res=1; while(n){ if(n&1) {res \*= a;--n;} else {a \*= a;n >>= 1;}} return res;}",

            "template <typename T> bool revsort(T a, T b){return a > b;}",

            "const int MAX\_N = 1e5 + 5;",

            "const ll MOD = 1e9 + 7;",

            "const ll INF = 1e20;",

            "const ld EPS = 1e-20;",

            "ll maxm(vector<ll> v){",

                "sort(all(v));",

                "return v[v.size()-1];",

            "}",

            "ll minm(vector<ll> v){",

                "sort(all(v));",

                "return v[0];",

            "}",

            "void solve() {",

                "ll n;",

                "cin>>n;",

                "ll l[n];",

                "string s;",

                "cin>>s;",

                "for(ll i=0;i< n;i++){",

                "cin>>l[i];",

                "}",

            "}",

            "int main() {",

                "ios\_base::sync\_with\_stdio(0);",

                "cin.tie(0); cout.tie(0);",

                "int n = 1;",

                "cin >> n;",

                "for (int t = 1; t <= n; t++) {",

                    "// cout << 'Case #' << t << ': ';",

                    "solve();",

                "}",

            "}",

        ],

        "description": "This is a boiler plate"

    }

}