{

    /"": {

        "prefix": "dfsbasic",

        "body": [

          "/\*dfs code: \*/",

          "const int M = 1e5+10;",

          "vector<int> g[M];",

          "bool vis[M];",

          "",

          "//4 sections in dfs to write code",

          "//every section has a definition",

          "void dfs(int vertex){",

          "    /\* Take action on vertex after entering the vertex \*/",

          "    //if(vis[vertex]) return; //this can be written if inside for we havent written vis child condn",

          "    cout<<vertex<<\"\\n\";",

          "    vis[vertex] = true;",

          "    for(int child:g[vertex]){",

          "        cout<<\"parent\"<<vertex<<\" ,child\"<<child<<\"\\n\";",

          "        if(vis[child]) continue;",

          "        /\* Take action on child before entering the child node \*/",

          "        dfs(child);",

          "        /\* Take action on child after existing the child node \*/",

          "    }",

          "    /\* Take action on child before existing the vertex \*/",

          "}"

        ],

        "description": ""

      }

}