{

"Default cpp formate": {

"prefix": [

"advanced",

],

"body": [

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

"using namespace std;",

"typedef long long ll;",

"#define pi (3.141592653589)",

"#define mod 1000000007",

"#define float double",

"#define pb push\_back",

"#define mp make\_pair",

"#define ff first",

"#define ss second",

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

"#define min3(a, b, c) min(c, min(a, b))",

"#define min4(a, b, c, d) min(d, min(c, min(a, b)))",

"#define rrep(i, n) for(int i=n-1;i>=0;i--)",

"#define rep(i,n) for(int i=0;i<n;i++)",

"#define fast ios\_base::sync\_with\_stdio(false), cin.tie(nullptr), cout.tie(nullptr);",

"\n",

"int32\_t main(){",

"fast",

"\n",

" $0",

"return 0;",

"}",

],

"description": "This is a c++ sniffpet",

},

"for loop": {

"prefix": "forl",

"body": ["for($1 $2 = $3 ; $2 < $4 ; $2++)", "{", " ${0:/\* code \*/}", "}"],

"description": "For Loop"

},

"Queries": {

"prefix": "query",

"body": [

"int t;",

"cin>>t;",

"while(t--)"

"{",

" $0",

"}",

],

"description": "q queries"

},

"primecheck": {

"prefix": "isPrime",

"body": [

"bool isPrime(int n)"

"{",

" if(n==1) return false;",

" if(n==2) return true;",

" for(int i=2;i\*i<=n;i++)"

" {",

" if(n%i==0)return false;",

" }",

" return true;",

"}\n",

],

"description": "to check n is prime or not"

},

"basic": {

"prefix": "basic",

"body": [

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

"using namespace std;",

"typedef long long ll;",

"#define mod 1000000007",

"#define rep(i,n) for(int i=0;i<n;i++)",

"#define fast ios\_base::sync\_with\_stdio(false), cin.tie(nullptr), cout.tie(nullptr);",

" ",

"int32\_t main()",

"{fast",

" $0",

" return 0;",

"}",

],

"description": "basic to start with"

},

"arrayofn": {

"prefix": "intarr",

"body": [

"int n;",

"cin>>n;",

"int arr[n];",

"for (int i=0; i<n; i++)cin>>arr[i];",

"$0"

],

"description": "int array of n size",

},

"ETF": {

"prefix": "ETF",

"body": [

"int phi(int n)",

"{",

" int res=n;",

" for(int i=2;i\*i<n;i++)",

" if(n%i==0)",

" {",

" res/=i;",

" res\*=(i-1);",

" while(n%i==0)n/=i;",

" }",

" "

" if(n>1)res/=n,res\*=(n-1);",

" return res;",

"}",

],

"description": "to find the number of coprime less than n",

},

"Vectorindex": {

"prefix": "vecindex",

"body": [

"int getIndex(vector<int> v, int K)",

"{",

" auto it = find(v.begin(), v.end(), K);",

" ",

" if (it != v.end())",

" {",

" int index = it - v.begin();",

" return index;",

" }",

" return 0;",

"}",

],

"description": "to find th index of element in vector",

},

"intrev": {

"prefix": "intrev",

"body": [

"ll intrev(ll a)",

"{",

" ll x=a;",

" vector<int>dgts;",

" ll ans=0;",

" while(x>0)",

" {",

" dgts.push\_back(x%10);",

" x/=10;",

" }",

" int n=dgts.size();",

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

" ans+=(pow(10,n-1-i)\*dgts[i]);",

" return ans;",

"}",

],

"description": "to get the reverse of an int",

},

}