1>exit(EXIT\_FAILURE)->terminate whole program not just block

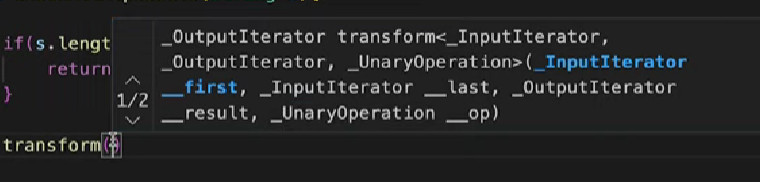
2>count(S.begin(),s.end(),s[i]): count occurrence of a character “s[i]” in a given a string “S”

#include<cctype>

3> toupper: char s=toupper(‘s’);

4> tolower

5>





6> #include<bits/stdc++.h>

using namespace std;

int main() {

int t;

cin>>t;

while(t--){

int n;

string b;

cin>>n>>b;

set<char> dist(b.begin(),b.end());

string r(dist.begin(),dist.end());

int l = r.length();

string s {};

for(int i =0;i<n;i++){

int index = lower\_bound(r.begin(),r.end(),b[i])-r.begin();

s.push\_back(r[l-index-1]);

}

cout<<s<<"\n";

}

}

**XOR:**

The XOR operator has some useful properties:

* a⊕a = 0 (any number XORed with itself is 0)
* a⊕0=a (any number XORed with 0 is the number itself)
* XOR is commutative and associative, meaning the order of operations does not matter

.

**Next\_permutation:**



#STRING :: NPOS

class Solution {

public:

    bool rotateString(string s, string goal) {

        if(s.size()!=goal.size()) return false;

        return (s+s).find(goal)!=string::npos;

    }

};