**4CS401: Cryptography and Network Security**

**B.Tech. (CSE) – I [ 2022-23 ]**

**Assignment**

**Prime Factorization**

**Title: Prime Factorization**

**Code :**

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

**#define ll long long**

**#define ul unsigned long long**

**#define pb emplace\_back**

**#define po pop\_back**

**#define vi vector<ll>**

**#define vii vector<vector<ll>>**

**using namespace std;**

**void file(){**

**ios\_base::sync\_with\_stdio(false);**

**cin.tie(NULL);}**

**ll M = 1e9 + 7;**

**string rsanum;**

**int rem;**

**string longDivision(string number, int divisor)**

**{**

**string ans;**

**int idx = 0;**

**int temp = number[idx] - '0';**

**while (temp < divisor)**

**temp = temp \* 10 + (number[++idx] - '0');**

**while (number.size() > idx) {**

**rem = temp % divisor;**

**ans += (temp / divisor) + '0';**

**temp = (temp % divisor) \* 10 + number[++idx] - '0';**

**}**

**// cout<<rem;**

**if (ans.length() == 0)**

**return "0";**

**if(rem==0)**

**return ans;**

**else return number;**

**}**

**void solve(){**

**string num;**

**rem=0;**

**cin>>num;**

**unordered\_map<int,int> mp;**

**rsanum = num;**

**int len = num.size();**

**int lastDigit = num[len-1] - '0';**

**// cout<<num;**

**string ans = longDivision(num,2);**

**// cout<<rem;**

**while(rem == 0){**

**mp[2]++;**

**num =  ans;**

**// cout<<ans;**

**ans = longDivision(num,2);**

**// cout<<rem<<" ";**

**}**

**for (int i = 3; i <= 1000000; i = i + 2)**

**{   string ans = longDivision(num,i);**

**while (ans!="0" && rem==0)**

**{   mp[i]++;**

**num =  ans;**

**ans = longDivision(num,i);**

**//cout<<rem<<" ";**

**}**

**}**

**cout<<"\n";**

**for(auto x:mp) cout<<x.first<<"-"<<x.second<<"\n";**

**}**

**int main()**

**{   file();**

**int t=1;**

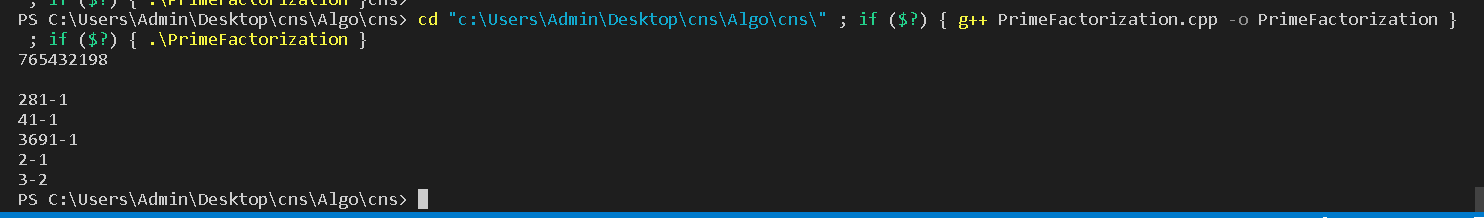
**while(t--)**

**solve();**

**return 0;**

**}**

**Output:**

****