

LAB 2 TASK 1:

```
message = 'odroboewscdrolocdcwkbdkmdzvkdpybwyeddrobo' #encrypted  
message
```

```
LETTERS = 'abcdefghijklmnopqrstuvwxyz'
```

```
for key in range(len(LETTERS)):
```

```
    translated = ''
```

```
    for symbol in message:
```

```
        if symbol in LETTERS:
```

```
            num = LETTERS.find(symbol)
```

```
            num = num - key
```

```
            if num < 0:
```

```
                num = num + len(LETTERS)
```

```
            translated = translated + LETTERS[num]
```

```
        else:
```

```
            translated = translated + symbol
```

```
    print('Hacking key #%%s: %%s' % (key, translated))
```

RESULT: ethereum is the best smart contract platform out there

LAB 2 TASK 2:

"af p xpkcaqvnkp pfg, af ipqe qpri, gauuikifc tpw, ceiri udvk tiki afgarxifrphni cd eaowvmd popkwn, hiqpvri du ear jvaql vfgikrcpfgafm du cei xkafqaxnir du xrwqedearcdkw pfg du ear aopmafpcasi xkdhafmr afcd fit pkipr. ac tpr qdoudkcafm cd lfdt cepc au pfwceafm epxxifig cd ringdf eaorinu hiudki cei opceiopcaqr du cei uaing qdvng hi qdoxnicinw tdklig dvcpgf edt rndtnw ac xkdqiigig, pfg edt odvfcparfvr cei dhrcpqnrceiki tdvng pc niprc kiopaf dfi mddg oafg cepc tdvng qdfcafvi cei kiripkqe"

aceah toz puvg vcdl omj puvg yudqecov, omj loj auum klu thmjuv hs klu zlcvu shv zcbkg guovz, upuv zcndu lcz vuwovroaeu jczyyuvomdu omj qmubyudkuj vukqvm. klu vcdluz lu loj avhqnlk aodr svhw lcz kvopuez loj mht audhwu o ehdoe eunumj, omj ck toz yhyqeoveg auecupuj, tlakupuv klu hej sher wcnlk zog, klok klu lcee ok aon umj toz sqee hs kqmmuez zkqssuj tckl kvuozquv. omj cs klok toz mhk umhqnl shv sowu, kluvu toz oezh lcz yvhehmnuj pcnhqv kh wovpue ok. kcwu thvu hm, aqk ck zuuwuj kh lopu eckkeu ussudk hm wv. aonncmz. ok mcmukg lu toz wqdl klu zowu oz ok scskg. ok mcmukg-mcmu klug aunom kh doee lcw tuee-yvuzuvpuj; aqk qmdlomnuj thqej lopu auum muovuv klu wovr. kluvu tuvu zhvu klok zlhhr klucv luojz omj klhqnlk klcz toz khh wqdl hs o nhhj klcmm; ck zuuwuj qmsocv klokomghmu zlhqej yhzuz (oyyovumkeg) yuvyukqoe ghqkl oz tuee oz (vuyqkujeg) cmubloqzkcaeu tuoekl. ck tcee lopu kh au yocj shv, klug zocj. ck czm'k mokqvoe, omj kvhqaue tcee dhvu hs ck! aqk zh sov kvhqaue loj mhk dhvu; omj oz wv. aonncmz toz numuvhqz tckl lcz whmug, whzk yuhyeu tuvu tceecmn kh shvncpu lcw lcz hjjckcuz omj lcz nhhj shvkqmu. lu vuwocmuj hm pczckcmn kuvwz tckl lcz vueokcpuz (ubduyk, hs dhqvzu, klu zodrpceeu- aonncmzuz), omj lu loj womg juphkuj ojwcvuvz owghmn klu lhaackz hs yhhv omj qmcwyhvkomp sowcecu. aqk lu loj mh dehzu svcumjz, qmkce zhvu hs lcz ghqmnuv dhqzcmz aunom kh nvht qy. klu uejuzk hs kluzu, omj aceah'z sophqvcku, toz ghqmnuv svhjh aonncmz. tlum aceah toz mcmukg-mcmu lu ojhykuj svhjh oz lcz lucv, omj avhqnlk lcw kh ecpu ok aon umj; omj klu lhyuz hs klu zodrpceeu- aonncmzuz tuvu scmoeej jozluj. aceah omj svhjh loyyumuj kh lopu klu zowu acvkljog, zuyuwauv 22mj. ghq loj aukku dhvu omj ecpu luvu, svhjh wg eo, zocj aceah hmu jog; omj klum tu dom dueuavoku hqv acvkljog-yovkcu dhvshvkoag khnuklu. ok klok kcwu svhjh toz zkcee cm lcz ktuumz, oz klu lhaackz doeeuj klu cvvuzymzcaeu ktumkcuz auktuom dlcejlhj omj dhwcmm hs onu ok klcvkg-klvuu

```
#include <iostream>
```

```
#include <map>
```

```
#include <algorithm>
```

```
#include <vector>
```

```
using namespace std;
```

```
int main() {
```

```
    map < char , int > mp;
```

```
    int total = 0;
```

```
    string s = "aceah toz puvg vcdl omj puvg yudqecov, omj loj auum klu thmjuv hs klu zlcvu  
shv zcbkg guovz, upuv zcndu lcz vuwovroaeu jczyyuvomdu omj qmubyudkuj vukqvm.  
klu vcdluz lu loj avhqnlk aodr svhw lcz kvopuez loj mht audhwu o ehdoe eunumj, omj ck toz  
yhyqeoveg auecupuj, tlokupuv klu hej sher wcnlk zog, klok klu lcee ok aon umj toz sqee hs  
kqmmuez zkgssuj tckl kvuozqvu. omj cs klok toz mhk umhqnl shv sowu, kluvu toz oezh lcz  
yvhehmnuj pcnhqv kh wovpue ok. kcwu thvu hm, aqk ck zuuwuj kh lopu eckkeu ussudk hm  
wv. aonncmz. ok mcmukg lu toz wqdl klu zowu oz ok scskg. ok mcmukg-mcmu klug aunom  
kh doee lcw tuee-yvuzuvpuj; aqk qmdlomnuj thqej lopu auum muovuv klu wovr. kluvu tuvu  
zhwu klok zlhhr klucv luojz omj klhqnlk klcz toz khh wqdl hs o nhhj klcmn; ck zuuwuj  
qmsocv klokomghmu zlhqej yhzuz (oyyovumkeg) yuvyukqoe ghqkl oz tuee oz (vuyqkujeg)  
cmubloqzkcaeu tuoekl. ck tcee lopu kh au yocj shv, klug zocj. ck czm'k mokqvoe, omj  
kvhqaeu tcee dhwu hs ck! aqk zh sov kvhqaeu loj mhk dhwu; omj oz wv. aonncmz toz  
numuvhqz tckl lcz whmug, whzk yuhyeu tuvu tceecmn kh shvncpu lcw lcz hjjckcuz omj lcz  
nhhj shvkqmu. lu vuwocmuj hm pczckcmn kuvwz tckl lcz vueokcpuz (ubduyk, hs dhqvzu,  
klu zodrpeeu- aonncmzuz), omj lu loj womg juphkuj ojwcvuvz owghmn klu lhaackz hs yhhv  
omj qmcwyhvkomp sowcecu. aqk lu loj mh dehzu svcumjz, qmkce zhwu hs lcz ghqmnuv  
dhqzcmz aunom kh nvht qy. klu uejuzk hs kluzu, omj aceah'z sophqvcku, toz ghqmn svhjh  
aonncmz. tlum aceah toz mcmukg-mcmu lu ojhykuj svhjh oz lcz lucv, omj avhqnlk lcw kh  
ecpu ok aon umj; omj klu lhyuz hs klu zodrpeeu- aonncmzuz tuvu scmoeeg jozluj. aceah  
omj svhjh loyyumuj kh lopu klu zowu acvkljog, zuykuwauv 22mj. ghq loj aukku dhwu omj
```

ecpu luvu, svjhj wg eoj, zocj aceah hmu jog; omj klum tu dom dueuavoku hqv acvkljog-yovkcuз dhwshvkoaeg khnucluv. ok klok kcwu svjhj toz zkcee cm lcz ktuumz, oz klu lhaackz doeuj klu cvvuzymzcaeu ktumkcuz auktuum dlcejlhj omj dhwcmm hs onu ok klcvkг-klvuu"

```
for(auto lol: s){
    if('a' <= lol && 'z' >= lol) mp[lol]++; total++;
    vector < pair < double , char > > vp;
    string ef = "etaonhisrdluwmgcfybpkvjxzq";
    string key = "abcdefghijklmnopqrstuvwxyz";
    string f = "";
    for(auto lol: mp) {
        vp.push_back({(lol.second * 100.00) / total, lol.first});
    }
    sort(vp.begin(), vp.end());
    for(auto lol: vp){
        //cout << lol.first << " " << lol.second << endl;
        f += lol.second;
    }
    reverse(f.begin(), f.end());
    //cout << "f : " << f << endl;
    int c = 0;
    for(auto lol: f){
        //cout << key[(int)ef[c] - (int)'a'] << " will be replaced as " << lol << endl;
        key[(int)ef[c] - (int)'a'] = lol;
        c++;
    }
}
```

```
cout << "Key Found : " << key << endl;
string decrypted = s;
int i = 0;
for(auto lol: s) {
    if('a' <= s[i] && 'z' >= s[i]) decrypted[i] = key[(int)lol - (int)'a'];
    i++;
}
cout << "AFTER DECRYPTION : " << endl;
cout << decrypted << endl;  return 0;}
```