Phill-CPP-0524

1. 處理較複雜的字串陣列

   aAbBc12345@#$% → 大寫:? 小寫, 數字, 特殊

```
#include <iostream>
#include <cctype>
using namespace std;

int main()
{
    string input;
    cin >> input;

    int a=0, b=0, c=0, d=0;

    for(size_t i=0; i<input.length(); i++){
      char single = input[i];

      if(isupper(single)){
        a++;
      }
      else if(islower(single)){
        b++;
      }
      else if(isdigit(single)){
        c++;
      }
      else{
        d++;
      }
    }
    cout << "uppercase ->"<< a<<endl;
    cout <<"lowercase -> "<< b<<endl;
    cout << "digit->" << c<<endl;
    cout << "others->" <<d << endl;


    return 0;
}
```

請挑出字元

請使用 c-style 字串陣列

```
#include <iostream>
#include <cstring>
#include <cctype>
using namespace std;

int main()
{
    char input[101];
    cin>>input; // C++ -> char array

    int len= strlen(input);

    char upper[101]="";
    char lower[101]="";
    char digits[101]="";
    char others[101]="";


    int upper_count=0, lower_count=0, digits_count=0, others_count=0;

    for(size_t i=0; i<len; i++){
      char single = input[i];

      if(isupper(single)){
        upper[upper_count++]= single; //生成字元陣列
      }
      else if(islower(single)){
        lower[lower_count++]=single;
      }
      else if(isdigit(single)){
        digits[digits_count++]=single;
      }
      else{
        others[others_count++]=single;
      }
    }
    cout << "uppercase ->"<< upper_count <<"->"<< upper <<endl;
    cout <<"lowercase -> "<< lower_count << "->" << lower<<endl;
    cout << "digit->" << digits_count <<"->" << digits<<endl;
    cout << "others->" <<others_count << "->"<< others << endl;


    return 0;
}
```

3.“Hello Advantek!”

→ 不管大小寫, 忽略digits others

→

H or h = ? 個

E or e = ? 個

……

```
```

4.兩個字串之間的關係

ABCDABBADCAB

AB

1 5 11

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string str1, str2;
    cin>> str1 >> str2;

    size_t str1_len = str1.length();
    size_t str2_len = str2.length();
    //ABCDABBADCAB
    //AB


    for(size_t i=0; i<=str1_len -str2_len; i++){//比差額
      if(str1.substr(i, str2_len)==str2){ //切片
          cout << i+1 << " ";
      }
    }


    return 0;
}
```