

預設以字元和數字之間隔開

```
請輸入要加密的key: ga12m3e  
切出來的字串: g a 12 m 3 e
```

如果有加 ' , ' 會用 ' , ' 優先隔開

```
請輸入要加密的key: ga12,3me  
切出來的字串: g a 12 3 m e
```

也可以輸入負數

```
請輸入要加密的key: ga-1,2,-3me  
切出來的字串: g a -1 2 -3 m e
```

Mod 26 的結果(括號)

```
請輸入要加密的key: ga-12,34,2me  
切出來的字串: g a -12(14) 34(8) 2(2) m e
```

組出來的 key

```
請輸入要加密的key: imitation-1,-2,-3game  
切出來的字串: i m i t a t i o n -1(25) -2(24) -3(23) g a m e  
imtaonzyxgebcdfhijklprsuwv
```

加密

```
您好，請問要加密還是解密？  
請輸入數字1(加密) or 2(解密):1  
請輸入key: ga-34,27,-2me  
切出來的字串: g a -34(18) 27(1) -2(24) m e  
key_str=gasbyme cdfhijklmnopqrtuvwxyz  
請輸入要加密的plaintext:  
abcd  
您的ciphertext是:  
gasb
```

解密

```
您好，請問要加密還是解密？
請輸入數字1(加密) or 2(解密):2
請輸入key: ga-34,27,-2me
切出來的字串: g a -34(18) 27(1) -2(24) m e
key_str=gasbymeCDFHIJklNopqrTUVWxz
請輸入要解密的ciphertext:
gasb
您的plaintext是:
abcd
```

其餘防呆同上一次作業

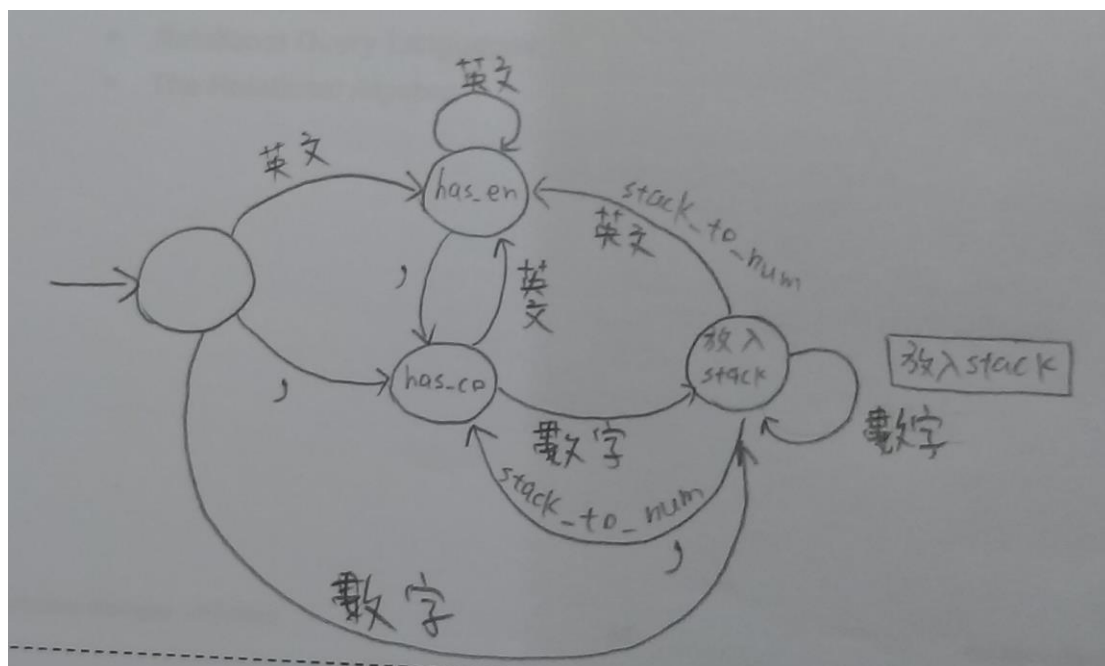
選擇加密或解密時的防呆(只要不是輸入 1 or 2 就不會繼續執行)

大小寫的防呆，大寫會自動轉乘小寫

除了英文字母一律不加密

程式碼

finite automaton



只要是數字一律先放入 stack

遇到英文字母或逗號就確認 stack 裡有沒有數字

如果有($n \neq -1$)就拿出來 \rightarrow stack_to_num

```

if('a'<=s[i] && s[i]<='z'){
    n=stack_to_num();//把stack存的數字取出
    if(n!=-1 && mark[n]==false){ //如果stack有數字的話
        key_str[c_i++]=w[n];
        mark[n]=true;
    }

    if(mark[s[i]-97]==false){
        key_str[c_i++]=s[i];
        mark[s[i]-97]=true;
    }

    printf("%c ",s[i]);
}

```

```

else if(s[i]==',' ){
    n=stack_to_num();
    printf("%d\n",n);
    if(n!=-1 && mark[n]==false){
        key_str[c_i++]=w[n];
        mark[n]=true;
    }
}

```

利用 mark 確認之前有沒有被加進去，如果沒有就加入 key_str

```

} else if('0'<=s[i] && s[i]<='9'){//放入stack
    stack[s_i++]=s[i]-48;//放入stack
    if(i==s.length()-1){//如果是最後一個字元
        n=stack_to_num();
        if(n!=-1 && mark[n]==false){
            key_str[c_i++]=w[n];
            mark[n]=true;
        }
    }
}
}

```

如果數字是最後一個字元也要被拿出來

```

int stack_to_num(){
    if(s_i==0) return -1;
    int num=0;
    for(int i=0;i<s_i;++i){
        num+=stack[i]*pow(10,s_i-i-1);
        // printf("%d\n",num);
    }
    if(has_neg){
        num=-num;
        has_neg=false;
    }
    printf("%d",num);
    num=num%26;
    if(num<0) num+=26;
    printf("( %d) ",num);
    s_i=0;
    return num;
}

```

負數轉換和 mod 都在這處理

```

for(int i=0;i<26;++i){
    if(mark[i]==false){
        mark[i]=true;
        key_str[c_i++]=w[i];
    }
}

```

最後把剩下的字母填入 key_str