1　暗号の解読

普通は平文 (P) の文字に鍵字 (K) の文字を加えて暗号文 (C) の文字にするので、これを、

$$P + K = C$$

と表現する。翻訳するには、逆に減法をして、

$$C - K = P$$

とする。あるいは、$26 - K$ に相当する補字、たとえば B に対して Z、C に対して Y、・・・を逆の鍵 (K') として、

$$C + K' = P$$

とする。

しかし、暗号文の作成に、加法ではなく

$$K - P = C$$

と減法を使う方式が一部の機械暗号で使われた。この減法方式では鍵字が A でも暗号になる。結果的には各文字は補字に写されて、それからいくつかずらされる。この方式が好都合なのは、こうすると翻訳も

$$K - C = P$$

であり、暗号文の作成とまったく同じ操作で可能な点である。

暗号の数理　一松　信　ブルーバックス　26 頁

1.1 暗号の解読 ango1.c

```
/*
  暗号の解読
  ango1.c
*/
#include <stdio.h>
int main()
{
  char table[] = {'a','b','c','d','e','f','g','h','i','j',
                  'k','l','m','n','o','p','q','r','s','t',
                  'u','v','w','x','y','z','?'};
  char *p;
  char ango[20];
  int  index;
  int  key;

  printf("key->");          //鍵字を入力
  scanf("%d", &key);
  printf("ango->");         //平文、暗号分を入力
  scanf("%s", ango);
  printf("%s\n", ango);     //入力した平文、暗号文を表示

  p = ango;
  while(*p != '\0'){
    if('a' <= *p && *p <= 'z'){
      index = key - (*p - 97);
      if(index < 0){
        index = index + 26;
      }
    }else{
      index = 26;
    }
    putchar(table[index]);
    p++;
  }
  return 0;
}
```