北一女中 2014
資訊選手培訓營
0818-0822

Agenda

- 8/18(一) 樹狀結構與圖論 (怡芬老師)
- 8/19(二) 排序演算法 (怡芬老師)
- 8/20(三) 網路概論 (致平老師)
- 8/21(四) 動態規畫法 (致平老師)
- 8/22(五) 資料壓縮與資料加密 (怡芬老師)

資料壓縮的目的

- **減少檔案佔有的儲存空間：**透過資料壓縮的方式可達到善用有限的儲存空間，並常常有將資料妥善整理的效果。
- **使檔案更好攜帶傳遞：**太大的檔案都將影響區域網路傳輸品質與網頁作品的呈現，透過資料壓縮的技巧將可改善這些問題，且壓縮軟體也有提供保密的功能，增加資料傳遞的安全性。

3

資料壓縮的種類

- **『無失真壓縮法』(Lossless compression)**
  - 重建結果和壓縮前的資料源完全相同，用於文字檔、執行檔、醫學影像和重要資料壓縮。
  - PCX 、GIF 、TIFF 、TGA、PNG 等影像格式，或者是ZIP，RAR等的資料壓縮都屬於此類
- **『破壞性壓縮法』(Lossy compression)**
  - 重建結果和壓縮前的資料源不完全相同，會發生資料短少的現象，常用於需要高壓縮比且允許部分失真的影像壓縮
  - JPEG、MPEG、聲音壓縮(如MP3)屬於此類

4

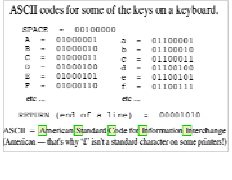
資料壓縮系統

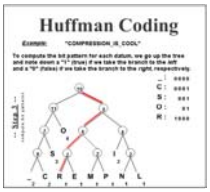
- **資料冗餘(data redundancy)：**為達到資料壓縮的目的，通常必須找出存在原始資料間多餘資料，減少並移除它，就能達到資料壓縮的目的。

- **編碼效率(coding rate，Cr)**
  Cr= (原始資料的大小 – 壓縮後的資料大小 / 原始資料大小) × 100 %

編碼（Coding)

- 編碼是將資料中所有的字母或是符號一一編成相對應且唯一的二進位序列（binary sequences）。
- 編碼種類：
  - **固定長度碼(fixed-length code)** 如：ASCII碼；
  - **變動長度碼（variable-length code）** 如：摩斯碼(Morse)與霍夫曼編碼（Huffman coding）等。





固定長度編碼

- 一般採用美國國家標準資訊交換碼(American Standard Code for Information Interchange , **ASCII**)來表示個人電腦所需要的文數資料。
- 每個符號的ASCII碼都由七個位元組合而成，由於每個位元可為1或0，故可用來表示128種不同的符號字元。由於每一個符號資料都使用固定長度的字碼來表示，此種編碼方式稱為**固定長度編碼**(fixed-length code)。
- 舉例說明，A～F以ASCII來表示如下：

符號	ASCII字碼 (7 bits)
A	1000001(65)
B	1000010(66)
C	1000011(67)
D	1000100(68)
E	1000101(69)
F	1000110(70)

變動長度編碼法（Run Length Code)

- **原理**：把資料中重複多次的內容，記錄其內容細節與出現的次數

Raster

9	9	6	6	6	6	6	7
6	6	6	6	6	6	6	6
9	9	6	6	6	6	7	7
9	8	9	6	6	7	7	5

Run-length codes

2:9, 5:6, 1:7
8:6
2:9, 4:6, 2:7
1:9, 1:8, 1:9, 2:6, 2:7, 1:5

Image Compression

Image

Pixel Values

0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 1 1 1 1 0 0
0 0 1 1 1 1 0 0
0 0 1 1 1 1 0 0
0 0 1 1 1 1 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

repeated values
= redundancy,
= opportunity
for compression

Raw pixel data:
0 0 0 0 0 0 0 0, 0 0 0 0 0 0 0 0, 0 0 1 1 1 1 0 0, 0 0 1 1 1 1 0 0, 0 0 1 1 1 1 0 0, 0 0 1 1 1 1 0 0, 0 0 0 0 0 0 0 0, 0 0 0 0 0 0 0 0.

Run-Length Encoded:
8(0), 8(0), 2(0) 4(1) 2(0), 2(0) 4(1) 2(0), 2(0) 4(1) 2(0), 2(0) 4(1) 2(0) ,8(0), 8(0).

Further Encoded:
2(8(0),) 4(2(0) 4(1) 2(0),) 2(8(0),).

Symmetry Encoded:
+(2(4(0)), 2(2(0) 2(1)),). "+" = four-fold symmetry

Run-length Compression
程度★　難度★

- 壓縮前：aaaabbcabcbbbaaaa
- 壓縮後：a4b2c1a1b1c1b3a4

- UVa 11541 12547

d021: 2007 程式達人
D - Run Length Encoding

Run Length Encoding (RLE)編碼方式是多媒體資料壓縮常用的方法之一，RLE 的作法是於將一連串相同的資料改以兩個部分來表示，前面一部分是資料本身(symbol)，後面部分代表該串資料的長度(也就是重複次數, run length)。例如:輸入字串為 "aaaabbcdeeeeeefghhhij"，經過 RLE 編碼後結果為"a4b2c1d1e5f1g1h3i1j1"。

當然，上面的例子其實沒有達到壓縮檔案的目的，有部分的原因是出現一次的 字元卻得用字元+次數(1)來表示，為了節省空間，有人提議出現一次的字元長度就無須紀錄，因此將表示方法改為: 字元+後續出現次數，例如 "aaaa"經過編碼後為" a3" (a 出現一次後又再出現三次)，因此輸入字串 "aaaabbcdeeeeeefghhhij"的編碼就變為:"a3b1cde4fgh2ij"。

但是上面表示方法仍有問題，因為 symbol 後有可能接的是symbol，也有可能是run-length，造成混淆，因此又有一個解決方案如下:若出現次數大於一，重複該字元兩次，並接上剩餘重複 次數，例如:"aaaa"經過編碼後為" aa2"，" bbb"經過編碼後為" bb1"，所以只要字元重複，表示後面接的是數字，若未重複，則該字元僅出現一次且其後也緊接另一個字元。依此原則，"aaaabbcdeeeeeefghhhij" 將被編碼為"aa2bb0cdee3fghh1ij"。請參照以上說明，編寫 RLE encoder。

輸入說明：輸入為多筆測試資料，每一行輸入皆由大小寫英文字母所組成的字串。此字串長度最長為1000 個字元。

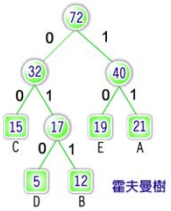
輸出說明：每一行輸出為相對應輸入字串編碼後的結果。

範例輸入：aaaabbcdeeeeeefghhhij

範例輸出：aa2bb0cdee3fghh1ij

霍夫曼編碼法(Huffman Encode)

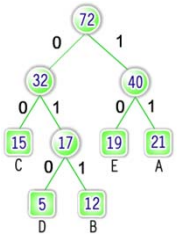
- **霍夫曼(Huffman)**在1952年所提出的一種無失真壓縮技術。
- 是一種不固定長度的編碼技術，符號的編碼長度與出現頻率成反比，屬於頻率相關編碼(frequency dependent encoding)。



第12頁

霍夫曼樹

- 原理：將要壓縮之字串，先讀一遍，再將字串中的每一個相異單字元的出現頻率，做成統計，依此來建構霍夫曼樹。
- 將左樹枝標0，右樹枝標1，即形成一個編碼樹。



第13頁

霍夫曼編碼法步驟

Step1：找出所有符號的出現頻率。
Step2：將出現頻率由小到大排列。
Step3：將頻率最低的兩者相加得出另一個頻率，頻率小的在左側。
Step4：重複步驟3，直到只剩下一個頻率為止。
Step5：將左樹枝標0，右樹枝標1，即形成一個編碼樹。
Step6：將各個符號的編碼做轉換。

第14頁

步驟1

- 找出所有符號的出現頻率。

例：AACADEDDEBCAABEDCADE

符號	A	B	C	D	E
頻率	6	2	3	5	4

第15頁

步驟2

- 將出現頻率由小到大排列。

符號	A	B	C	D	E
頻率	6	2	3	5	4

符號	B	C	E	D	A
頻率	2	3	4	5	6

第16頁

步驟3

- 將頻率最低的兩者相加得出另一個頻率，頻率小的在左側。

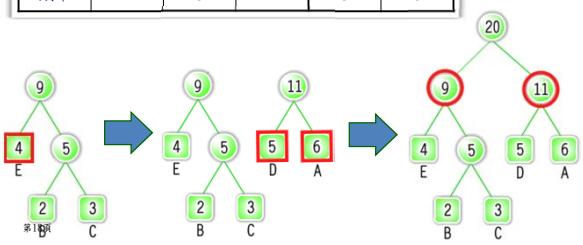
符號	B	C	E	D	A
頻率	2	3	4	5	6

第17頁

步驟4

- 重複步驟3，直到只剩下一個頻率為止。

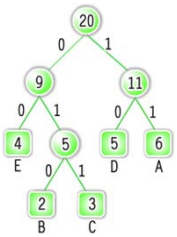
符號	B	C	E	D	A
頻率	2	3	4	5	6



第18頁

步驟5

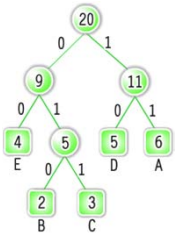
- 將左樹枝標0，右樹枝標1，即形成一個編碼樹。



第19頁

步驟6

- 寫下各個符號的霍夫曼碼。

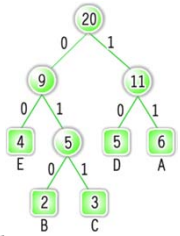


A：11
B：010
C：011
D：10
E：00

第20頁

Huffman 編碼

- 透過下表，我們可以將
  AACADEDDEBCAABEDCADE轉換成
  11110111110001010000100111111010001001111 1000



A：11
B：010
C：011
D：10
E：00

第21頁

霍夫曼編碼法的應用

- 常常被拿來處理大量的符號編寫工作。
- 根據整組資料中符號出現的頻率高低，決定要給這個符號怎麼編碼
- 如果符號出現的頻率太高，則給符號的碼越短，相反符號的號碼越長。
- 常常用來處理資料壓縮的問題。

第22頁

Huffman Code 練習

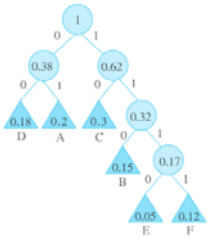
- 假設編碼系統中A、B、C、D、E等五個符號，其出現頻率依序為0.15、0.35、0.12、0.1、0.28，請據此設計一套霍夫曼碼。

(1) 假設編碼系統中有 A、B、C、D、E、F 等符號，其出現頻率依序為 0.2、0.15、0.3、0.18、0.05、0.12，請據此畫出編碼樹並設計一套霍夫曼碼。

(2) 依照上一題所設計的霍夫曼碼，將 111110010000110 進行解碼。

解答：

(1)



A：01
B：110
C：10
D：00
E：1110
F：1111

(2) FCADDB

MP3音樂壓縮原理

何謂MP3

- **MPEG-1 Audio Layer 3**，經常稱為MP3，是目前較流行的一種數位音訊編碼和有損壓縮格式。
- 用來大幅度地降低音訊資料量，而對於大多數使用者來說，重放的音質與最初的不壓縮音訊相比沒有明顯的下降。
- **特性**：檔案小、音質佳、容易分享，壓縮比可高達1：12。
- **原理**：將人耳所能聽到範圍以外的音域去除後，再使用霍夫曼編碼法(Huffman Encode)加以壓縮。

數位影像與壓縮技術

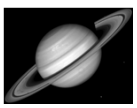
數位圖形與影像的描述

- **向量圖(vector based image)**：圖形(Graphics)
  - 用一種命令來描述圖形。
  - 繪圖程式(draw program)：AutoCAD、Corel Draw、3DS MAX
  - 檔案小
  - 工程製圖、廣告設計、電腦輔助設計(CAD)、地圖等
- **點陣圖(bit-mapped image)**：影像(image)
  - 描述畫面中每一個圖素的亮度或顏色。
  - 畫圖程式(paint program)：小畫家、Photo Shop
  - 影像檔：檔案大，PCX、BMP、TIFF、TGA、GIF

28

影像分類

- 黑白(二元化)影像 (Binary images)
- 灰階影像 (Gray-scale images)
- 彩色影像 (Color images)




- 靜態與動態影像



點陣圖重要參數

- **影像尺寸**：影像的大小，例如320*200
- **彩色空間**：顏色描述的方法（或稱顏色模型），有RGB（紅綠藍）、CMYK（青澄黃黑）、YUV、YIQ。
- **位元平面(bit plane)**：一個位元所展開的平面。若(R,G,B)=(3,3,2)，則紅色就佔3個位元平面。
- **最大顏色數**：彩色影像各分量(R、G、B)所佔的位元平面數總和的位元空間。若(R,G,B)=(3,3,2)，則有256色。
- **影像資料容量**：影像所佔的空間（以byte為單位）=影像寬度*影像長度*影像深度/8

常用的向量圖檔格式

- **WMF**：windows metafiles，可嵌入點陣圖檔。
- **DRW**：micrografx公司所建立的格式，內含數學運算式。
- **CDR**：Corel Systems公司的Corel Draw所使用的格式。
- **EPS**：Encapsulated Post Script file，Adobe Systems公司所開發的格式，可包含點陣圖檔。

常用的點陣圖檔格式

- **BMP**：Microsoft windows BMP， Microsoft windows 的原始檔。
- **PCX**：Z-soft公司定義的格式，採用RLE無失真壓縮方法。
- **GIF**：Graphics Interchange Format，Compuserve公司開發的格式，使用LZW無失真壓縮技術，一個GIF檔可以有多幅彩色影像。
- **TIF**：Tag Image File Format，Aldus和Microsoft公司為掃描器所定的格式。

影像壓縮—依照壓縮方法分類

	無損壓縮	略損壓縮
壓縮後還原效果	保持原貌	略有不同
限制	不得失真	寬鬆
壓縮效果	有限	比無損壓縮好很多
適用範圍	文數字、程式等資料	影像、視訊與聲音等媒體

影像格式- BMP

- 微軟公司所提出的點陣圖格式
- 原本專門用在 Windows 作業系統
- 支援 RGB 全彩、索引色、灰階及黑白等色彩類型

影像格式- GIF

- 網頁上最常用的圖形格式
- 將原始影像資料中重複區塊編碼，然後再利用此代碼(索引值)來取代原始影像資料，每個索引值對應到一個影像區塊
- 有壓縮的效果
- 可以存成透明圖、交錯圖、和動畫，且提供「非破壞性壓縮」
- 存檔後的體積小，圖片不失真
- 缺點：最多只能儲存 256 色的色彩數

影像格式- PNG

- 採**非破壞性壓縮**，減少檔案的體積時也保有影像原本的品質
- 用來取代 GIF 格式，結合了 JPG 與 GIF 的優點
- 主要應用於 Internet 上
- 可存成交錯圖、透明圖
- 支援的色彩類型有：RGB 全彩、索引色、灰階及黑白模式

影像格式- TIF

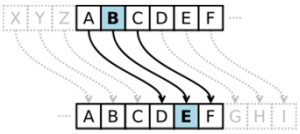
- 影像處理界普遍支援的圖檔格式
- 可以跨平台
- 提供非破壞性壓縮
- 適用於印刷輸出
- 大多數的影像處理軟體及排版軟體都會支援 TIF 圖檔

資料加密
Data encryption

資料來源：交大資工
蘇俊銘 (Jun Ming Su)

凱撒密碼法

- 在密碼學中，凱撒密碼（或稱凱撒加密、凱撒變換、變換加密）是一種最簡單且最廣為人知的加密技術。
- 它是一種替換加密的技術，明文中的所有字母都在字母表上向後（或向前）按照一個固定數目進行偏移後被替換成密文。



凱撒密碼法

- 凱撒密碼的加密、解密方法還能夠通過**同餘**的數學方法進行計算。
- 首先將字母用數字代替，A=0，B=1，...，Z=25。
- 此時偏移量為 n 的加密方法即為：

$$E_n(x) = (x + n) \mod 26$$

- 解密就是：

$$D_n(x) = (x - n) \mod 26$$

凱撒密碼法

加密： E(P)＝ (P+K) mod 26 ＝ C （mod是同餘數）

其中 E 是加密函數 P 是明文的字母 K 是右移量 C 是密文後字母

例如： K 取 3 可以產生下表 明文的字母

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

密文後字母

D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

利用凱撒密碼法對

"AS YOU SOW, SO SHALL YOU REAP. " 加密，會產生

"DV BRX VRZ, VR VKDOO BRX UHDS."

Q458: The Decoder

在密碼學裡面有一種很簡單的加密方式，就是把明碼的每個字元加上某一個整數K而得到密碼的字元（明碼及密碼字元一定都在ASCII碼中可列印的範圍內）。例如若K=2，那麼apple經過加密後就變成crrng了。解密則是反過來做。這個問題是給你一個密碼字串，請你依照上述的解密方式輸出明碼。

至於在本任務中K到底是多少，請自行參照Sample Input及Sample Output推出來吧！相當簡單的。

Input

每筆測試資料一列。每列有1個字串，就是需要解密的明碼。

Output

對每一測試資料，請輸出解密後的密碼。

Sample Input

1JKJ'pz'{ol'{yhklthyr'vm'{ol'Jvu{yvs'Kh{h'Jvywvyh{pvu5
1PTI'pz'h'{yhklthyr'vm'{ol'Pu{lyuh{pvuhs'I|zpulzz'Thjopul'Jvywvyh{pvu5
1KLJ'pz'{ol'{yhklthyr'vm'{ol'Kpnp{hs'Lx|pwtlu{'Jvywvyh{pvu5

Sample Output

*CDC is the trademark of the Control Data Corporation.
*IBM is a trademark of the International Business Machine Corporation.
*DEC is the trademark of the Digital Equipment Corporation.

Q10222: Decode the Mad man

很久以前，在波蘭的BUET大學有一個老教授完全發瘋了。他開始用很特殊的文字來說話。沒有人可以聽得懂他的演講及授課。最後，BUET終於陷入了難題。再也沒有方法可以讓他繼續在大學裡工作了。突然有一個學生（當然是UVA ACM學會註冊的作者，並在24小時線上裁判系統上有很好的排名）創造了一個可以將教授的話解碼的程式。在此之後，這個老教授開始一如往常般地工作，每個人也都鬆了一口氣。因此，如果你有機會去到BUET，看到一個老師透過一個接到裝有語音辨識的IBM電腦的麥克風說話，而學生則從電腦螢幕上聽課，你可別嚇到！因為現在你的工就是要寫一個一樣可以解碼這個瘋教授語言的程式。

輸入檔僅含一筆測試資料，也就是編碼後的訊息。這筆測試資料含有一個或多個單字。

輸出：就所給的測試資料，將解碼後的單字印在一行。還好，這工作並不難，只要把每個字母或符號以鍵盤上在它左邊第二個鍵的符號來取代就行了。

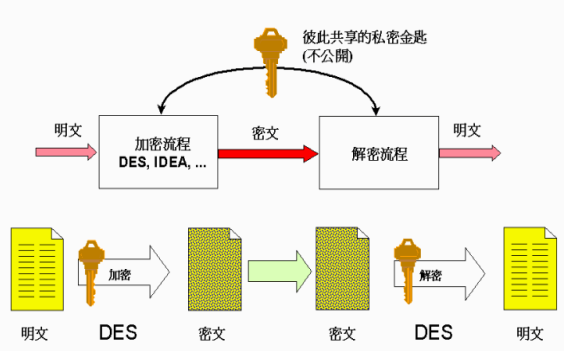
範例輸入
k[r dyt i[o
p "[nt]y[jyd.

範例輸出
how are you
i love program

加密技術體系

- **對稱式加密體系(Symmetric Key)**
  - 加密金鑰和解密金鑰相同
  - 對稱演算法速度快，所以在處理大量資料的時候被廣泛使用，其關鍵是保證密鑰的安全。
  - 最具影響的是DES密碼
- **對稱式金鑰(Asymmetric Key)體系**
  - 分別存在一個公鑰和私鑰，公鑰公開，私鑰保密。
  - 公鑰和私鑰具有一一對應的關係，用公鑰加密的資料只有用私鑰才能解開。
  - 其最有影響的公鑰加密演算法是RSA。

對稱式加密技術



DES (Data Encryption Standard)簡介

- **基本原理：**利用Shannon的多重加密的觀念，並利用Confusion（混淆）與Diffusion（散佈）等方式：將明文(PlainText)轉換成其他格式，並散佈明文的每一個小部分擴散到密文的各部分以達到加密效果。
- **保密技巧：**將原始資料「明文」弄得非常散亂，讓破解者無法利用統計方式或其他數學分析技巧將加密後的「密文(Ciphertext)」還原成原來的明文。
- **加密方法：**透過16回合的運算(位移)所組成，每一回合的運算目的，在於將上一回合所打散的明文在弄得更亂一些。就是指每一次的運算相當於在明文中多加了一道鎖，因此經過DES運算之後，其原始資料已被16把鎖給保護住了。

46

RSA加密演算法

- 西元1997，三位美國麻省理工學院學者李瓦士（Rivest）、夏米爾（Shamir）、以及艾道曼（Adleman）率先公開RSA加密演算法並取得專利權
- 此演算法是最先進及最方便的加密方法，它在電子商務交易中扮演了相當重要的角色
- 目前有很多的數位消費性產品，例如視訊轉換器與智慧卡，都是利用了RSA加密來傳遞訊息
- RSA就是分別取三位學者名字的開頭字母來命名的。

http://eportfolio.lib.ksu.edu.tw/~4960E085/wiki/index.php/RSA%E5%8A%A0%E5%AF%86%E6%BC%94%E7%AE%97%E6%B3%95

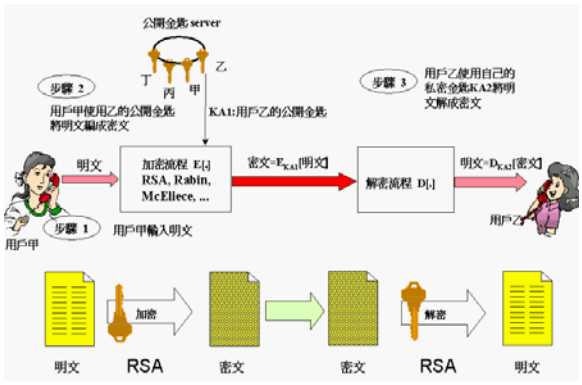
RSA加密演算法

- RSA加密 演算法是一種特殊的非對稱密碼法，**利用兩個質數作為加密與解密的兩個鑰匙(key)**。這兩個鑰匙分別稱為公開鑰匙 (public key) 和私人鑰匙 (private key 或是 secret key)，**鑰匙的長度約在 40 個位元到 1024 位元**。
- 公開鑰匙作為加密，只有使用私人鑰匙才能解密，解密者只要不洩露私人鑰匙，別人就算有公開鑰匙，也是很難推演算出來私人鑰匙，就算是利用反向工程來解密也不是一件簡單的事，
- 所以 RSA 算是一種十分安全的加密與解密演算法。

RSA加密演算法

- 如果你想和別人祕密通訊，那麼你可以先選定兩個非常巨大的質數P_1、P_2作為私人鑰匙（(private key，解密用的)），然後將$P_1 \times P_2$的乘積作為加密用的公開鑰匙(public key)，你可以把公開鑰匙(public key)公佈在名片上或在網路上。
- 那麼，別人要傳一封密函給你，他必需要先得到你的公開鑰匙(public key)，按照一個約定的方法將信件加密後送出。你在收到密函後，再用你的私人鑰匙(private key)就可以解出密函原文。

RSA運作方式



RSA演算法 (非對稱式加密技術)

- RSA基於數學難題，具有**大數因數分解**。
- RSA使用兩個密鑰：公鑰(Public Key, PK)與私鑰(Private Key, SK)
  - 加密時把明文分成塊，塊的大小可變，但不超過密鑰的長度。
  - RSA把明文塊轉化為與密鑰長度相同的密文

RSA 運作方式

1. 假設資料(Data)要由A機器傳至B機器：
   - 由B機器用亂數決定一個key,我們稱之為私鑰Private Key (SK) 。
   - 這個key都只留在B機器裡不送出來。
2. 由這個SK計算出另一個key
   - 稱之為公開金鑰Public Key (PK) 。
   - 這個Public Key的特性是幾乎不可能反演算出Private Key
   - 然後將這個Public Key透過網路丟給A機器.
3. A機器將資料(明文)用這個Public Key編碼：
   - 這個編碼過的資料(密文)一定得使用Private Key才解得開。
   - 然後A機器將密文透過網路傳給B機器。
4. B再用Private Key將資料解碼。

如果有第三者竊聽資料時：
- 他只能得到B傳給A的Public Key，以及A用這個Public Key編碼後的密文。
- 沒有SK，第三者無法解碼，所以RSA方法確實能防止第三者的竊聽。

- **RSA的安全性依賴於大數分解：**
  - 公鑰和私鑰都是兩個大質數（大於 100 個十進制位）的函數。
  - 從一個密鑰(SK)和密文推斷出明文的難度等同於分解兩個大質數的積。
  - **密鑰對的產生：**
    - 選擇兩個大質數：p 和q 。
    - 計算： $n = p \times q$
    - 然後隨機選擇**加密密鑰 e**，
      - 要求 e 和 $(p-1)\times(q-1)$ **互質**，即GCD(e, (p - 1) × (q - 1))=1。
    - 最後，利用 Euclid 算法計算**解密密鑰** d：
      - 滿足： $e \times d = 1 \bmod ((p-1)\times(q-1))$
      - 其中n和d也要**互質**。
      - 數e和 n是**公鑰**，d是**私鑰**。
      - 兩個質數p和q不再需要，應該丟棄，不要讓任何人知道。
  - **加密信息 m（二進制表示）時：**
    - 首先把m分成等長數據塊 m1, m2, ..., mi ，塊長s ，其中s≦ n， s 盡可能的大。
      - 0, 1, 2,..., 25與a, b, c,..., z一對一對應，把訊息中的文字轉變成數字M：
        - 例如YES→M=24q^2+4q^1+18q^0。
      - 對應的**密文**是：
        - $c_i = (m_i^{\ e}) \bmod n$ ..... (a)
      - **解密**時作如下計算：
        - $m_i = (c_i^{\ d}) \bmod n$ ..... (b)
  - **RSA 可用於數位簽章(Digital Signature)：**
    - 用 (a) 式簽名，用 (b)式驗證。操作時考慮到安全性和 m信息量較大等因素，一般是先作 HASH(雜湊) 運算。

RSA與DES之比較

	RSA	DES
發表年代	1977	1976
發明人	美國麻省理工學院三位教授 Rivest、 Shamir、 Adlemam	IBM 及美國國家安全局
基本特徵	加密 Key ≠解密 Key	加密 KEY = 解密 Key
主要優點	Public Key可以公開，而且可以提供數位簽章	加密解密速度快
主要缺點	解密速度慢，Key 生成耗時初期系統成本高	Key 傳送困難並且 Key 必須共享

現今的加密技術

- 要破解128位元對稱加密的數位金鑰
  - 需要花上自現在至太陽再度循環為新星、吞噬地球所需的時間。
- 要破解1024位元非對稱加密的數位金鑰
  - 亦需要相同的時間。
- 比較可能的危險
  - 可能因不小心遺失了金鑰密碼提示句而使您的金鑰遭到盜取的機率，要遠高於金鑰遭到破解的機率。

55

SSL(Secure Socket Layer)

- SSL 是由 Netscape 所發展，它是介於 Application Protocol 和 TCP/IP 間一個公開的、公用的資料安全性之通訊協定。
- SSL 之功能有為傳輸資料加密、連結伺服器之認證、確保傳送信息之完整性等。
- 即為與另一方在通訊之前先講好的一套方法。
  - 此方法可以在通訊雙方之間建立一個秘密通道。
  - 凡是一些不希望被他人知道的機密資料都可以透過此通道傳送給對方
  - 這樣一來即使資料必須要通過公開通路(如 Internet)，也不用擔心資料會被別人偷窺。

Factorization

因式分解。這裡談的是把一個正整數分解成質因數的連乘積： 例如：20 = 2^2 * 5

當要因式分解的數字有很多筆時，可以先建好質數表，然後只拿質因數來試除。

UVa 516 583 10179 10290 10329 10392 10622 10780 10791

Trial Division Factorization Method

把所有可能的因數拿來試除。用質因數會更好。

```
void trial_division(int n)
{
  // 窮舉n所有可能的因數一一試除。
  // 最簡單的方式是找由2到n當中的數字。
  for (int d=2; d<=n; ++d)
    while (n % d == 0)
    {
      n / = d;
      cout << d; // 印出質因數
    }
}
```

Trial Division Factorization Method

```
void trial_division(int n)
{
  // 一個數字n不會有大於sqrt(n)的質因數（除了n本身以外）
  int sqrt_n = sqrt(n);
  for (int d=2; d<=sqrt_n; ++d)
    while (n % d == 0)
    {
      n / = d;
      cout << d; // 印出質因數
    }

  if (n > 1) cout << n;  // n是質數
}
```

a010-質因數分解

http://zerojudge.tw/ShowProblem?problemid=a010

因數分解就是把一個數字，切分為數個質數的乘積，如 12=2^2 * 3　　其中, 次方的符號以 ^ 來表示

輸入說明 ：一個整數, 大於1 且 小於等於 1000000

輸出說明 ：一個字串

範例輸入：若題目沒有特別說明，則應該以多測資的方式讀取

```
20
17
999997
```

範例輸出：

```
2^2 * 5
17
757 * 1321
```