


Grace
Hopper
柔雞曰：她頗正：“>”

"A ship in port is safe, but that's not what ships are built for."

在港口的船是安全的，但那不是船被建造的目的。



我小時候最喜歡數學課
熱愛研究量角器計算尺

我是世界最早一批電腦
程式設計師之一

當時還是我用髮夾把
繼電器裡面的bug夾出
來的。

我是在**紐約市**出生的



我畢業於 Vassar 大學

我的退休的 party 上，
總共來了兩千多個人

我的博士論文是寫：
用一個幾何的方法去
解決一個代數的問題

1966年我滿六十歲，
海軍又叫她退休了。



1941年，歐
戰爆發，讓
我有了從軍
的念頭

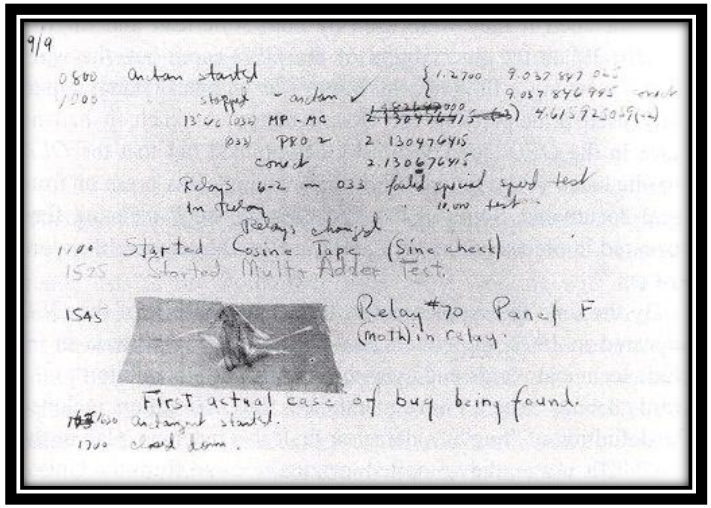
歐戰爆發，
我被分派到
哈佛大學的
一個後勤的
計算局

小時候我喜歡玩各種的器具，量
角尺、計算尺，我都喜歡拿來玩



1945.09.09 A BUG

- 這天發生了一樁對電腦界而言非常重要的軼事。那天因為天氣很熱，所以實驗室裡的窗戶都開著，也不知道什麼時候飛進來一隻蛾。結果那隻蛾被打死在繼電器裡面，造成電路不通，讓機器當機，使他們沒辦法算出他們要的結果。經過了近一天的檢查，Hopper 找到了那隻蛾，並她用她的髮夾去把那隻蛾給弄出來，而且他還把那隻蛾的屍體貼在她的管理日誌上，上面寫著：「**就是這個 bug，害我們今天的工作無法完成。**」這個消息傳開之後，那個實驗室裡的人每逢老闆詢問為何還沒做出結果時，都把**過錯推給 bug**。
- 在這件事情之後，在**電腦界中遇到程式中有錯就稱之為 bug**。除錯叫做 debug。






那為什麼她可以在歷史上名留青史阿?
當然是因為他發明了偉大的編譯器阿
是Grace Hopper欸
呃…他是誰?
就一個偉人阿~
喔
喔拜託!這是一件超屌的事欸

電腦應該要是個麻雀雖小五臟俱全的東西



- Grace Hopper：「在古老的時候，人都是用牛跟馬來拖東西，後來人類發展越來越蓬勃，東西越來越重，但是人們沒有發明更大更壯的物種來搬運東西；事實上，人們是用更大一批的牛跟馬來做這些更加粗重的工作。」
- 以這個比喻來說，電腦不應該越做越大，應該**用小小的東西，但是用一大堆來做要做的事情。**

編譯器的出現

早期，要把**我們的語言**變成**電腦的語言**是要靠研發人員在紙上打洞然後放到機器裡面去讀取並且翻譯

Grace進入Eckert-Mauchley公司之後，他說他希望可以有一種程式可以直接把想做的事情寫下來，然後這個程式可以直接把英文翻譯成“電腦語言”

所有人都告訴她電腦只能做計算，只能加減乘除，只能處理數字， 電腦是不懂英文的。

1952 年 Grace 終於發展出第一套實驗性的編譯器， 在 Sperry 的機器上可以執行的，叫做 A-0，後來又發展了 A-1、 A-2。 。

讓我們繼續看下去

由此可見，**1950年代**，大部分**人尚未意識到電腦是用來處理資訊的工具**，但所有人只認為**電腦是個計算機**



當初在寫 A-0 時，為了向出錢的老闆炫耀，她還設計了三個版本：英文、德文、法文。她想證明 compiler 不僅可以看得懂英文，也看得懂德文跟法文。但是後來德文跟法文的 compiler 都沒再做下去。

後來 Grace Hopper 漸漸發展她的 compiler 在 UNIVAC I & II 上面能夠執行， **1956 年她已經做出來一套蠻完整的程式語言，叫做 FLOW-MATIC**

對電腦界的第二大貢獻

FLOW-MATIC 這套系統在Grace 還是海軍的時候被**海軍相中並開始使用**，後來海軍跟一些民間的公司想要把 FLOW-MATIC 變得更標準一點。



編譯器的結構

- 編譯器是在高層次的語言與底層硬件的節目源。
- 編譯器的”**生成物**”和”**反應物**”：
  1. 確定的**語法的正確性**的方案
  2. 產生正確和有效的**目標代碼**
  3. 運行時的組織
  4. 格式化輸出根據彙編器或鏈接器公約
- **三個部分組成**：**前端**、**中端**和**後端**編譯器



前端、中端、後端



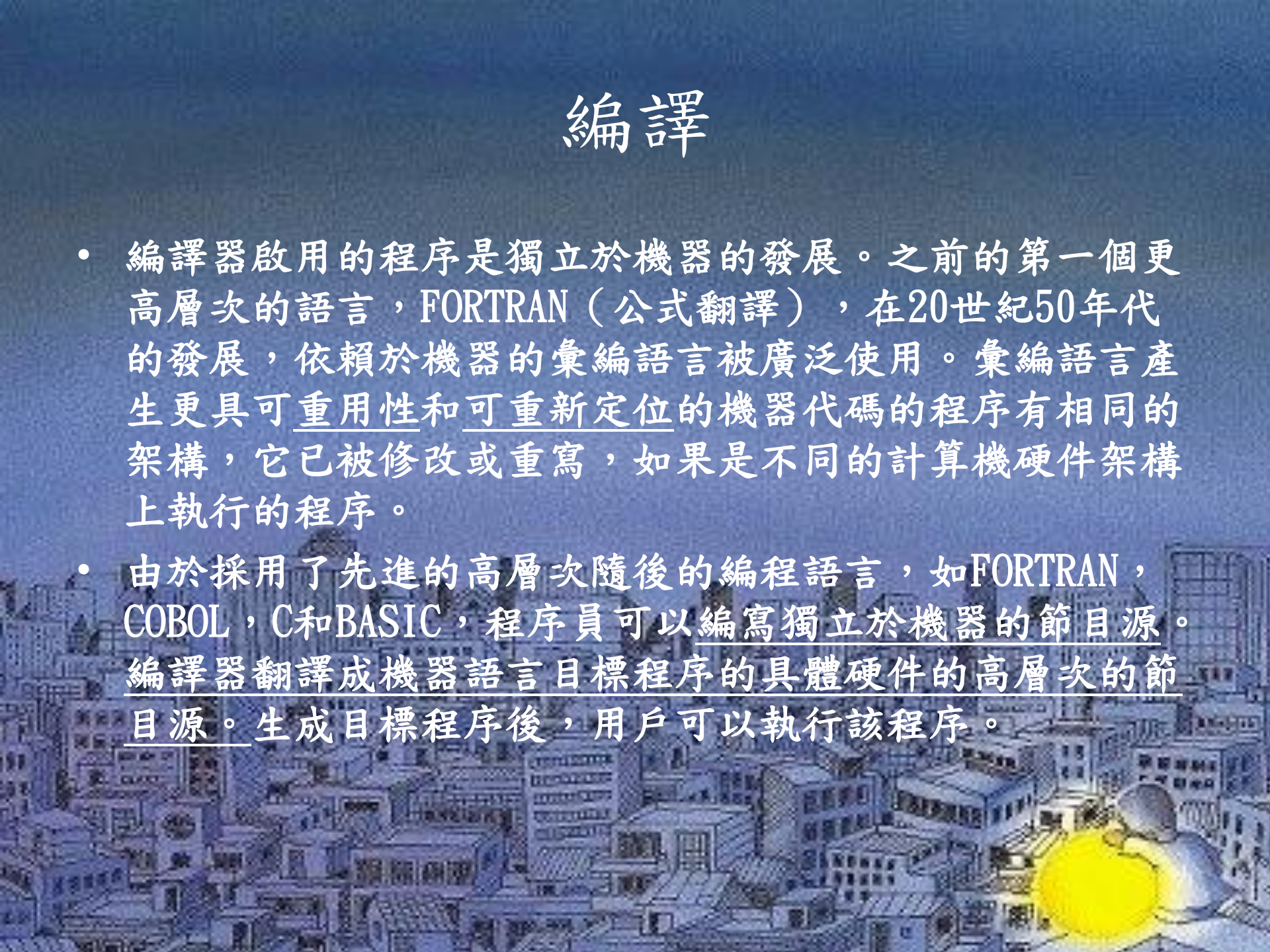
- **前端**檢查程序**是否被正確寫入的編程語言的語法和語義**。**這裡進行程序確認**。報告是否錯誤，如果有的話，即開始檢查也進行收集的訊息接著到前端處理的中端，然後**生成一個中間表示IR的源代碼**。

- **中端**優化發生於此。典型的優化是**除去無用或無法訪問的代碼**，發現和傳播常數值，計算搬遷到不經常執行的地方（例如，一個循環），或專業化上下文的計算。中端生成另一個IR。

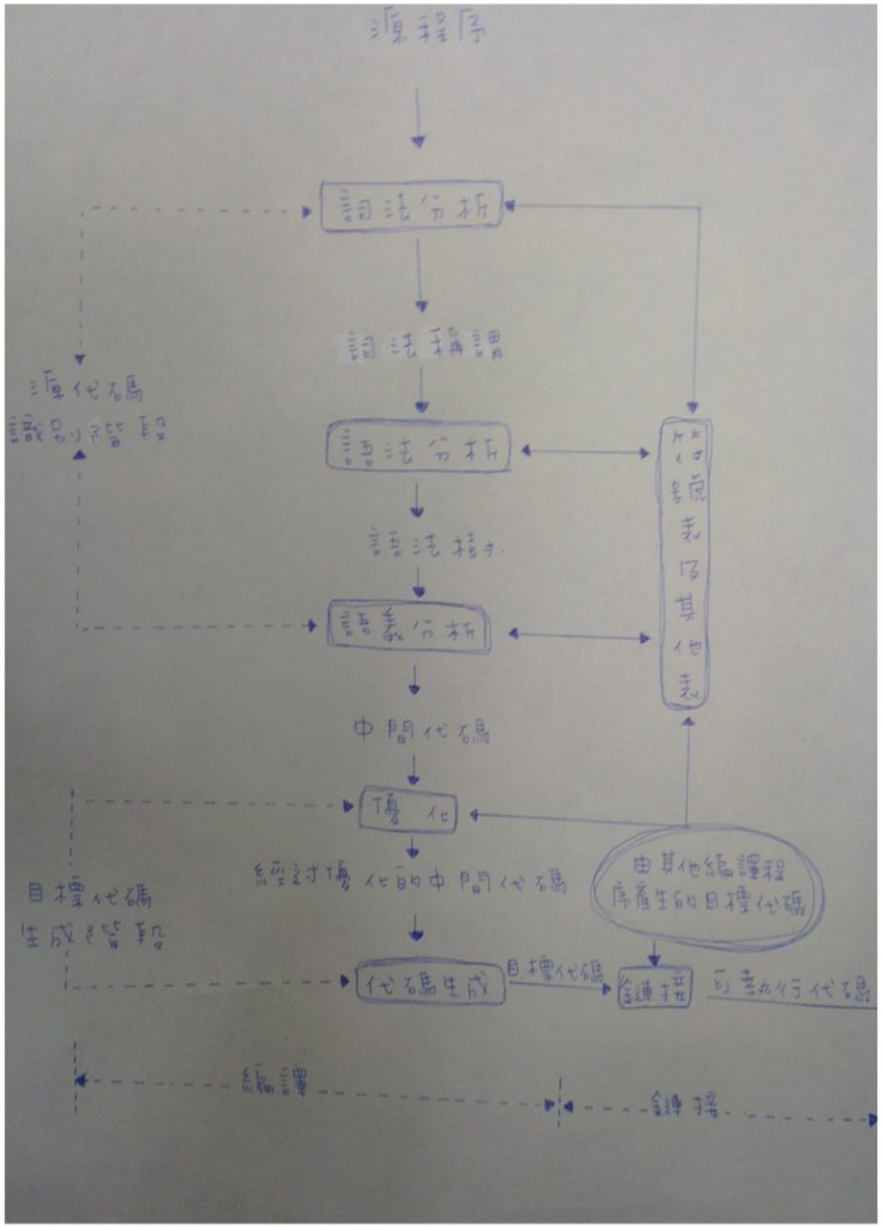
- **後端**是負責**翻譯成彙編代碼中端的IR**，最優化的努力都集中在這部分。目標指令（s）是選擇為每個IR指令寄存器分配受讓人處理器寄存器為在可能的情況下的程序變量。後端採用的硬件搞清楚如何並行執行單元保持忙碌，填充延遲槽，等等。雖然NP最優化算法，啟發式技術發達。

編譯

- 編譯器啟用的程序是獨立於機器的發展。之前的第一個更高層次的語言，FORTRAN（公式翻譯），在20世紀50年代的發展，依賴於機器的彙編語言被廣泛使用。彙編語言產生更具可重用性和可重新定位的機器代碼的程序有相同的架構，它已被修改或重寫，如果是不同的計算機硬件架構上執行的程序。
- 由於採用了先進的高層次隨後的編程語言，如FORTRAN，COBOL，C和BASIC，程序員可以編寫獨立於機器的節目源。編譯器翻譯成機器語言目標程序的具體硬件的高層次的節目源。生成目標程序後，用戶可以執行該程序。



編譯器的工作流程圖


源程序
詞法分析
詞法稱謂
源代碼識別階段
語法分析
語法樹
語義分析
符號表及其他表
中間代碼
優化
經過優化的中間代碼
由其他編譯程序產生的目標代碼
目標代碼生成階段
代碼生成
目標代碼
鏈接
可直接執行代碼
編譯
鏈接

編譯與解釋型語言

- **更高級的編程語言通常會使用一種類型的翻譯記住**：無論是設計，彙編語言或解釋型語言。然而，在實踐中有很少是任何一種語言，它需要專門編譯或專門解釋，雖然這是可能的設計依賴於在運行時重新解釋的語言。通常的分類反映了最流行 或最廣泛的一種語言的實現-。

- **解釋並不能完全取代編譯**。它只隱藏來自用戶的，並使得它逐漸。即使一個解釋器本身就可以解釋，直接執行的程序需要在堆棧的底部（機器語言）的地方。現代的趨勢剛剛在即時編譯和字節碼解釋有時模糊了傳統的分類已編譯器和解釋器。



編譯器完成初期的缺陷

- 缺陷：因為**海軍幅員廣大**，所以導致**各地自行稍將compiler作修改**，以致發生A地的程式在B地可能無法執行的問題。
- 解決方法：Grace Hopper寫了一套程式，用來**檢查這些程式之間是不是用同樣的編譯方式**， 稱作 Validation。後來影響到民間，漸漸發展出一套新的語言， 偏向商業用的語法，這套程式語言就是著名的 COBOL



COBOL

- 在1959年春天的CODASYL一個為期兩天的會議匯集了行業和政府的計算機專家。料斗擔任技術顧問委員會，和她的許多前僱員在短期委員會，**確定了新的語言**COBOL。新的語言**擴展Hopper的FLOW-MATIC語言的與IBM當量**，COMTRAN一些想法。Hopper的信念，程序應該被寫在一個語言，是接近到英國，而不是機器代碼或接近機器代碼（如彙編語言）被捕獲的新商業語言的語言，和COBOL會去上最普遍的商業語言的日期。
- 從1967年到1977年，料斗的海軍編程語言集團在海軍的信息系統規劃辦公室主任和被晉升職級的隊長在1973年。她開發的**驗證軟件COBOL**和它的**編譯器**的一部分，**整個海軍一個的COBOL標準化程序**。





如何證明她的重要性?







首次退休



- 1966 年 Grace Hopper 滿六十歲，海軍又叫她退休了(她當時的官階是海軍的中校，六十歲就該屆齡退休了)。 在退休的那一天，她說這是她一生中最悲傷的一天了， 因為人家已經告訴她她太老了。可是海軍在她退休後沒多久，發現沒有她是不行的，因為海軍裡還有很多電腦化、自動化、正規化的工作尚未完成，所以在她退休的五個月後，海軍就要求她復職。 所以在她退休後的半年她又回到了海軍。



- 回去之後擔任了好幾個海軍的**計算中心的主持人或是顧問**，指揮那些海軍的技術人員，到處去演講、宣傳她的觀念、管理觀。

二度退休

- 有一位參議員，他從來都沒見過 Grace。但他聽說了這個人以後，私下做了點調查，然後在國會裡面主動提出 Hopper 的申請案。他認為這樣重要的女性不應該只在海軍裡面作一個校官而已，覺得她應該做將官。在國會裡面，很快的通過這個提案，並且通知了海軍，海軍就將 Hopper 升職了。1958年11月，她當上了**海軍少將**，是**美國第一個女性到達這個官階的人**。1986 年，她七十九歲，是所有活著的、最老的、還沒除役的美國軍官。 此時是她第二度要退休。



退休後的生活

- 她退休之後，又跑去跟 Digital 公司簽約，Digital 公司聘請她代表他們演講、 推銷他們公司的產品。Digital 公司當時主要的產品是 minicomputer。 事實上，在 1986 年 PC 已經出來了，那時候叫做 microcomputer。 在海軍裡，Grace Hopper 是極力推展使用 PC 的人之一。



為什麼它/她會成功?

- 在他研發的過程中，為了籌足經費所以到處跟朋友借錢，有很多她的朋友問她說：「妳怎麼這麼勇敢呢？萬一搞砸了，妳就會賠上很多錢欸？」，Grace Hopper 回答道：「It ‘s always easier to ask forgiveness than it is to get permission. 」“它總是更容易請求寬恕比獲得許可。“
- 因為這句話，讓人看出了Grace為什麼會成功的原因，是因為他的心態。她有一種別人沒有的毅力。**成功與失敗只是一線之隔，成功沒有祕訣，成功並不是任何人的專利；成功只在於你有沒有那個意力堅持到最後。堅持到最後的人總比半途而廢的人來的有機會。**

名言佳句

- **Humans are allergic to change. They love to say, We have always done it this way. I try to fight that. That's why I have a clock on my wall that runs counter-clockwise.**

  人類對於改變很敏感。他們總是說：我們一直都是這麼做的。我嘗試挑戰這種想法。這也是為何我的時鐘是逆時鐘轉的。

- Leadership is a two way street, loyalty up and loyalty down. Respect for one's superiors; care for one's crew.

  領導統馭是雙向車道。對上忠誠也要對下忠誠。尊重上級，關懷部屬。

- One accurate measurement is worth a thousand expert opinions.

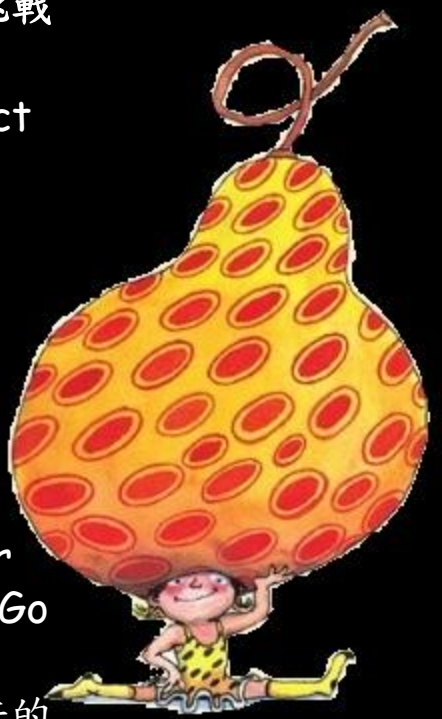
  一次精準的測量價值等同一千筆專家意見。

- To me programming is more than an important practical art. It is also a gigantic undertaking in the foundations of knowledge.

  對我來說程式設計不只是實用的藝術，也是建構知識基礎的巨大任務。

- At any given moment, there is always a line representing what your boss will believe. If you step over it, you will not get your budget. Go as close to that line as you can.

  任何情況下，你的老闆對你的信任總會有限度。如果你超過他對你信任的程度，將得不到一毛預算可用。在這範圍內盡你所能的逼近信任的極限。



榮譽

- 1969：榮獲首屆計算機科學的數據處理管理協會的年度最佳女子
- **1973：第一位美國女性成為了英國電腦學會資深會員。**
- 1986：收到美國國防部傑出服務獎章。
- 1987：獲得計算機歷史博物館研究員獎。
- 1988：在華盛頓特區獲得金槌獎演講會的國際公約。
- 1991：獲得了國家科技獎章。
- **2009：國家能源研究科學計算中心部命名了它的旗艦系統“漏斗”。**



心得報告

- 做完這次的報告之後又讓我認識了一個歷史上的偉人。原本拿到Grace Hopper這個題目後，本來還在想說從來沒聽過的這個人會做出什麼偉大的事情啊？結果開始蒐集資料後，才發現他所發明的編譯器原來是一個對電腦以及人類史上的重要發明。

心得報告

- 經過這次的報告，讓我認識了grace hopper這一個偉大的人!!也更了解她的一生及對這世界的貢獻。因為有她的發明，讓我們的電腦更方便使用，使我們的生活更便利。而在她的一生中，不管遇到什麼困難，卻從來不半途而廢的精神，值得我們學習。也讓我們學到，一個人要成功，是需要這麼大的努力。

心得報告

- 以前對Grace Hopper是多麼不了解，經這次的報告，我才明白，原來一個人要如此的成功，需要很多心血、精神，更重要的是要堅持到底不可半途而廢，每個人都可以成為像他一樣那麼成功的人，只是自己願不願意成為跟他一樣。但他還是讓我欽佩不已，更讓我了解只要努力每個人都有機會的。



參考資料

- http://en.wikipedia.org/wiki/Compiler
- http://en.wikipedia.org/wiki/Grace_Hopper




THE END