

### \$0.- Chuẩn bị làm việc với Turbo C

Bước 1. Copy tệp tc2-min.exe vào ổ C:\ rồi cho chạy giải nén thành thư mục TC2-MIN

Bước 2. Nên đổi tên thư mục thành TC, tức là bây giờ TC ở gốc của đĩa C:

Bước 3. Vào thư mục C:\TC chạy tệp tc.exe

Bước 4. Menu đang đen sẵn, chọn đang đen sẵn, gõ Enter, Enter và Enter để mở tệp TEST1.EXE đang có sẵn

Bước 5. Gõ F9 để dịch thử thấy báo lỗi, gõ Enter và Esc.

Gõ Alt+O và phím ↓ ↓ ↓ đến Directories, gõ Enter Enter Enter

Bước 6. Vào mục Include Directories thấy C:\TC2-MIN hiện đen sẵn

Gõ C:\TC thì màu đen biến mất và Enter.

Gõ ↓ và thực hiện như Bước 5. Cứ như vậy, để chữa hết các đường dẫn thành C:\TC.

Cuối cùng gõ Esc và Esc để thoát khỏi các công việc điều chỉnh.

Bước 7. Gõ Ctrl+F9 để chạy thử... Chạy xong gõ Alt+X để thoát.

### \$1.- Giới thiệu các menu quan trọng trong cửa sổ soạn thảo

Muốn soạn thảo một chương trình mới ta gõ tổ hợp phím Alt+F,N (tức là chọn New của menu File). Tên ban đầu là NONAME.C, và vùng soạn thảo hiện ra để bắt đầu...

Trái lại, muốn mở tệp cũ, ta gõ F3,↵, dùng các phím mũi tên dịch chuyển đến tệp cần load rồi gõ ↵, và vùng soạn thảo với nội dung của tệp đó hiện ra để bắt đầu...

Sau khi soạn xong, dịch thử bằng phím F9, chạy thử bằng Ctrl+F9, ta lưu tệp văn bản bằng phím F2. Nếu là tệp mới thì ta sẽ phải gõ tên tệp (không trùng với mọi tệp cũ trên cùng thư mục) rồi gõ ↵. Chú ý: Nếu viết đúng cú pháp mà vẫn còn lỗi: "Unable to open the File XXX.h", thì gõ tổ hợp phím Alt+O,D (tức là chọn Directories của menu Options), lặp công việc sau: (Gõ ↵, đặt lại đường dẫn đến thư mục hiện thời, ↵, ↓). Cuối cùng gõ Esc, Alt+O,S (tức là chọn Save của menu Options), ↵ để lưu các tùy chọn này vào thư mục TCCONFIG.TC dùng mãi về sau.

Trong khi soạn thảo ta có thể tạm thời trở về môi trường DOS bằng cách gõ tổ hợp phím Alt+F,O (tức là chọn OS Shell của menu File). Làm việc xong với DOS, ta quay về Turbo C bằng cách gõ lệnh exit tại dấu nhắc của DOS. Để kết thúc phiên làm việc với Turbo C, ta gõ tổ hợp phím Alt+F,X (tức là chọn Quit của menu File) hay phím nóng Alt+X. Nếu trong môi trường Windows ta phải gõ thêm Ctrl+C nữa.

Trong TC2, ta không dùng được con chuột như trong TP7. Mặt khác, thanh Menu của TC2, lại không có menu Help, việc tra cứu các mẫu lệnh sẽ khó khăn hơn. Menu Edit của TC2 nói chung không phát huy tác dụng. Muốn có một môi trường TC hoàn hảo, bạn nên sử dụng TC++ 3.0, trong đó có đủ Help, Edit và con chuột. Muốn vậy, hãy download từ trên mạng xuống hoặc cần 6 đĩa mềm.

### \$2.- Các phím soạn thảo gần giống như trong TP: **Xin nhắc lại một số thao tác chính là:**

**Home/End** Về đầu dòng. Đến cuối dòng.

**Ctrl+PageUp/Ctrl+PageDown:** Về đầu dòng đầu văn bản. Về sau kí tự cuối cùng văn bản.

**Ctrl+Y** Xoá dòng chứa con trỏ.

**Ctrl+Q,Y** Xoá từ vị trí con trỏ đến hết dòng.

**Ctrl+Q,A** Tìm kiếm/thay thế. Find (tìm gì?):... Replace (thay thế bằng?):....Options (tùy chọn): gõ ví dụ  
G (global), W (nguyên từ), ... và ↵. Tại mỗi chỗ, máy hỏi (Y/N) thì gõ ... và ↵.

**Tab** Căn lề trái: Đẩy dòng từ vị trí con trỏ sang phải vài vị trí. Nhưng nếu dòng trên không rỗng, nó sẽ đẩy tới vị trí thẳng cột với kí tự khác trống đầu tiên sau kí tự cách.

**Ctrl+K,B** Đánh dấu đầu khối

**Ctrl+K,K** Đánh dấu cuối khối

**Ctrl+K,C** Copy khối sao cho đầu khối vào đúng vị trí con trỏ

**Ctrl+K,V** Chuyển khối đi tới vị trí mới, mà đầu khối đúng vaog vị trí con trỏ.

**Ctrl+K,U** Dịch khối sang trái một kí tự.

**Ctrl+K,I** Dịch khối sangphải một kí tự.

**Ctrl+K,H** Ẩn/Hiện khối đã bị đánh dấu gần đây nhất.

**Ctrl+K,Y** Xóa khối đang được đánh dấu.

**Ctrl+K,W** Save khối vào một tệp trên đĩa mà ta sẽ gõ đường dẫn và tên tệp đầy đủ vào và ↵

**Ctrl+K,R** Viết ra màn hình tại vị trí con trỏ hiện thời một khối lấy từ một tệp có sẵn trên đĩa mà ta

sẽ phải gõ đường dẫn và tên tệp đầy đủ vào và gõ ↵.

**Alt+E, C** Copy khối đã đánh dấu vào ClipBoard (bộ nhớ đệm).

**Alt+E, P** Paste nội dung của ClipBoard vào văn bản đang soạn thảo.

### **Chú ý:**

Các tổ hợp phím **Alt+<số>** để mở cửa sổ thứ <số> không có tác dụng gì. Không có các động tác thu nhỏ, phóng to hay xếp chồng các văn bản như trong TP, vì chỉ có 1 cửa sổ làm việc Được hiện ra.

Các menu khác sẽ học dần hoặc bạn tự khám phá thử nghiệm thêm.

### **\$3.- Đặc điểm quan trọng nhất của ngôn ngữ C**

Tác giả ngôn ngữ C đã làm việc nhiều với ngôn ngữ máy, nên Dennis Ritchie luôn tạo ra các lệnh liên quan tới tận các bit. Ông này không chịu tham khảo phương pháp định kiểu/cỡ dút khoát từ trước như Pascal nên gây phiền toái cho người lập trình.

**Khi nhập/xuất dữ liệu ông ta mới cho định số lượng bit cho dữ liệu.**

#### **Ví dụ:**

Cùng là số 1 nhưng nếu là byte thì chiếm 8 bit, là word thì chiếm 2 bit,... Rồi kiểu có dấu và không dấu cũng được đặt ra. Chẳng hạn: số 129 cỡ 8 bit mà là số có dấu thì lại là -127, vì  $129 = 10000001$  và  $-127 = 1111111$ , cộng lại = 0.

Quy tắc đối với số có dấu và không dấu cỡ 8 bit như sau:

Với số không dấu thì chỉ cần đổi số dạng BIN này sang dạng DEC

Với số có dấu thì phải lấy bit cao nhất làm cờ đầu: 0 tức là dương, 1 tức là âm. Ví dụ, thấy 10000001 ta biết đây là số âm, vì bit bên trái là 1. Nhưng không có nghĩa là -1. Vì khi cộng với 1 nó lại bằng 10000010, nó lại = -2 mất, mà là 1111111, vì khi đó tổng của chúng moia bằng 100000000. Bit thứ 9 bị thừa ra, sẽ được cò tràn ghi nhận.

Công thức:  $A+B=0$  tương đương với  $A = \text{not}(B)+1$ , và  $B = \text{not}(A)+1$ .

#### **Ví dụ:**

Số nhị phân 10001001 biểu diễn số nguyên có dấu là bao nhiêu?

Ta thấy nó là một số âm, giả sử là -B. Thì  $B = \text{not}(A)+1 = 01110110+1 = 01110111 = 119$ .

Tức là nếu A là kiểu không dấu thì nó là 137, có dấu thì là -119. (Tổng của chúng =256).

Tương tự với số có dấu cỡ lớn hơn...

Chính vì lý do trên mà Tác giả luôn lưu tâm người lập trình phải khai báo rõ là số không dấu hay không.

### **\$4. Nhập xuất dữ liệu:**

a) **Mẫu nhập dữ liệu số:**

```
int m;  
float x;  
/* cách 1 */  
printf("Vao cac so: m,x=");  
scanf("%d%f*c",&m,&x); /* có thể bỏ *c */  
/* cách 2 */  
printf("Vao so: m=");  
scanf("%d",&m);
```

```
printf("Vao x=");  
scanf("%f",&x);
```

b) Mẫu định dạng:

`%d` số nguyên cỡ 2B

`%u` số tự nhiên (số nguyên không dấu) cỡ 2B

`%ld` số nguyên dài cỡ 4B

`%lu` số tự nhiên (số nguyên không dấu) cỡ 4B

`%c` kí tự

`%s` xâu kí tự

`%f` số thực

`%lf` số thực dài độ chính xác cao

`.*c` bỏ kí tự thừa ví dụ Enter

`%e` số thực dạng khoa học viết chữ e thường

`%E` số thực dạng khoa học viết chữ E HOA

`%o` số nguyên dạng OCT

`%x` số nguyên dạng HEXA, in thường

`%X` số nguyên dạng HEXA, IN HOA

**Các phép toán số học:** + - \* / (nếu hai toán hạng kiểu nguyên thì kết quả kiểu nguyên, giống DIV của TP)  
`%` (như MOD trong TP).

**Các phép toán quan hệ:** >, <, >=, <=, == (giống = của TP), != (giống <> của TP). Chúng cho kết quả đúng = 1 và sai = 0.

**Các phép toán logic:** && (như AND của TP), || (như OR của TP). Kết quả đúng = 1 và sai = 0.

**Lệnh gán:** <biến> = <biểu thức>;

Chú ý:

Đối với các phép + - \* / % có thể dùng câu lệnh tắt như ví dụ sau: `a+ = b;` tức là `a = a + b;` `a* = b;` tức là `a = a*b;`, ...

**Biểu thức gán:** (<biến> = <biểu thức>) có giá trị = kết quả của biểu thức và có kiểu của biến, không có dấu chấm phẩy.

Ví dụ:

`a=b=c=d=10;` hay có thể viết là `a=(b=(c=(d=10)))`;

**Biểu thức điều kiện:** có dạng `(e1?e2:e3)` cho cả biểu thức đó có giá trị bằng e2 nếu e1 khác 0, trái lại bằng e3, kiểu kết quả là kiểu cao nhất của e2 và e3.

Ví dụ:

`s=a>b?a:b;` cho s là max (a,b), `a>b` là một mệnh đề có giá trị bằng số: nếu `a>b` thì mệnh đề đó =1 trái lại mệnh đề đó = 0. Trong ví dụ này ta hiểu: nếu `a>b` đúng thì cho `s=a` trái lại thì cho `s=b`, hóa ra s là số lớn nhất trong 2 số a và b.

**Trong khi gán, thì giá trị về phải chuyển thành giá trị có kiểu của vế trái.**

Ví dụ: `int n; float f; x=n` là đúng, chẳng hạn `n=12` thì `f=12.0`.

**Trong biểu thức có nhiều kiểu khác nhau thì kiểu có giá trị thấp được tự động nâng lên thành kiểu cao nhất.**

Ví dụ:

`11/3` cho kết quả là 3, ở đây máy hiểu 11 là số kiểu nguyên. Còn `11.0/3` thì tự động thành `11.0/3.0` cho kết quả là 3.666666666 vì máy thấy mình viết 11.0 thì hiểu ngay 11 là số kiểu thực theo ý người lập trình. Cũng còn cách khác bắt nó cho kết quả kiểu khác như sau:

**Ép kiểu** theo nhu cầu của người lập trình: (<kiểu>)<biểu thức>. Khi đó biểu thức phải lấy kiểu đã xác định trong dấu ngoặc.

Ví dụ:

`int a,b; f=(float)a/b;` cho f giá trị thực. Nếu không ép kiểu thì kết quả là `a div b` mà thôi. Cũng vậy: `f=(float)11/3` cho kết quả `f = 3.666666666`.

Ví dụ:

`float r; n=(int)(r+0.5);` cho kết quả n là làm tròn lên của số thực r. Chẳng hạn với `r=3.4` thì `r+0.5=3.9` mà `(int)3.9=3`. Với `r=3.6` thì `r+0.5=4.1` và `(int)4.1=4`.

## §5. Cấu trúc cơ bản chương trình nguồn trong C

- Các chỉ thị cho bộ Tiền xử lý :

```
#include <tệp *.h>
```

```
#define <hằng> <giá trị hằng> /* Chú ý: tên hằng thường viết IN HOA */
```

- Định nghĩa kiểu:

```
typedefs <kiểu cũ> <kiểu mới> /* Chú ý: tên kiểu mới thường viết in hoa */
```

- Thân chương trình:

```
main() /* hoặc void main(void) hoặc void main() */
```

```
{
```

```
    Khai báo các biến, hằng...
```

```
    Các câu lệnh...
```

```
}
```

- Lần lượt là các chương trình con đều dưới dạng hàm như sau:

```
<kiểu giá trị hàm> <tên hàm>(kiểu giá trị tham số):<tên tham số>
```

```
{
```

```
    Khai báo các biến, hằng...
```

```
    Các câu lệnh...
```

```
}
```

## Bộ kí tự được phép dùng trong C:

52 chữ cái tiếng Anh

10 chữ số Ảrập

+ - \* / % ( ) [ ] { } < > ? ! , . : ; ` " ~ \_ \ | # \$ &

Các từ khoá đều viết chữ in thường: ví dụ: `clrscr();`

Các kiểu (so sánh với Pascal): (Không có kiểu Boolean trong C, có `Bool` trong C++)

C	Pascal	size (byte)	Khoảng giá trị
<code>int, short int</code>	<code>integer</code>	2	-32768 -> 32767
<code>long int</code>	<code>LongInt</code>	4	-2147483648 -> 214748647
<code>unsigned int</code>	<code>Word</code>	2	0 -> 65535
<code>unsigned long int</code>		4	0 -> 4294967295
<code>char</code>	<code>char, string</code>	1	0 -> 255
<code>float</code>	<code>real</code>	4	$3.4 \times 10^{-38}$ -> $3.4 \times 10^{38}$
<code>double</code>	<code>extended</code>	8	$1.7 \times 10^{-308}$ -> $1.7 \times 10^{308}$
<code>long float</code>		10	3.4E-4932->3.4E4932

Hằng là giá trị không đổi trong suốt chương trình.

**Chú ý:** Với các Hằng số nguyên:

Nếu tận cùng `L` hoặc `l` thì đó là kiểu long, ví dụ: `123L` hoặc `0.123E-12L`.

Tương tự:

U unsigned integer 50000U  
UL unsigned long 12345678UL  
F float 3E5F  
FL long float -4576FL

Nếu tiếp đầu là 0 thì đó là số **Hệ bát phân (OCT)**, ví dụ: 011 = 9.

Nếu tiếp đầu là 0X hay 0x thì đó là số **Hệ thập lục phân (HEX)**, ví dụ: 0X1A = 26, 0x1f = 31.

Hằng kí tự có thể tham gia vào các phép tính, ví dụ: '9'-'1' = 57 - 49 = 8.

Hằng kí tự có cách viết đặc biệt:

"\n" xuống dòng	"\?" dấu hỏi ?
"\a" chuông	"\r" về đầu dòng
"\b" lùi xoá	"\" vạch xiên xuôi
"\t" Tab ngang	"\%" dấu phần trăm.
"\" nháy kép	

Turbo C coi xâu kí tự là **Mảng các kí tự mà kết thúc bằng kí tự 0.**

Ví dụ: Khai báo `char ten[20];` tương đương với `var ten : string[20];` trong Pascal.

### Chú ý:

**Hằng xâu kí tự** phải được kẹp giữa hai dấu nháy kép, còn

**Hằng kí tự** trong dấu nháy đơn, có xâu rỗng, nhưng không có kí tự rỗng mà có kí tự cách.

Biến là giá trị có thể thay đổi trong chương trình.

Biến được khai báo <kiểu>:<ds tên biến>;

Ví dụ: `int x,y,z; float a,b,c;`

Có thể khai báo biến với giá trị ban đầu: ví dụ: `int x=100;`

Chú ý lệnh rẽ nhánh: `if` (biểu thức) Công việc 1; `else` Công việc 2;

Để dùng căn bậc hai `sqrt()`, thì phải thêm tệp bao hàm toán hoac: `#include <math.h>`

### Bài tập 1.

Giải phương trình bậc nhất, bậc 2 và hệ phương trình bậc nhất 2 ẩn, 3 ẩn.

### Bài tập 2.

Giải tam giác (tức là tính các yếu tố khác của tam giác: chu vi, diện tích, đường cao,...) theo a/ 3 cạnh a,b,c. Chú ý việc kiểm tra sự tồn tại của tam giác để có 3 cạnh là a, b và c).

b/ 3 cặp tọa độ các đỉnh trong mặt phẳng. Chú ý tới điều kiện không cho 3 điểm A, B và C thẳng hàng hay trùng nhau!.