

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ  
TRƯỜNG CAO HỌC CÔNG NGHỆ  
KHOA ĐIỆN

# KỸ THUẬT LẬP TRÌNH C

GIÁO TRÌNH THỰC HÀNH

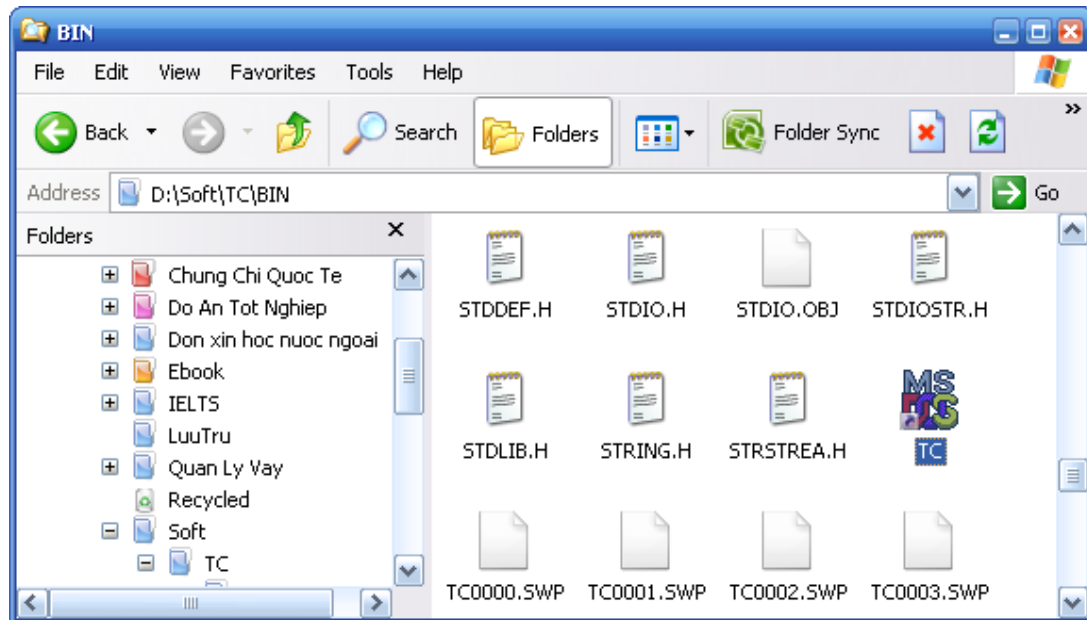
Giảng viên: NGUYỄN THỊ HÀ QUYÊN

Hà Nội, 2009

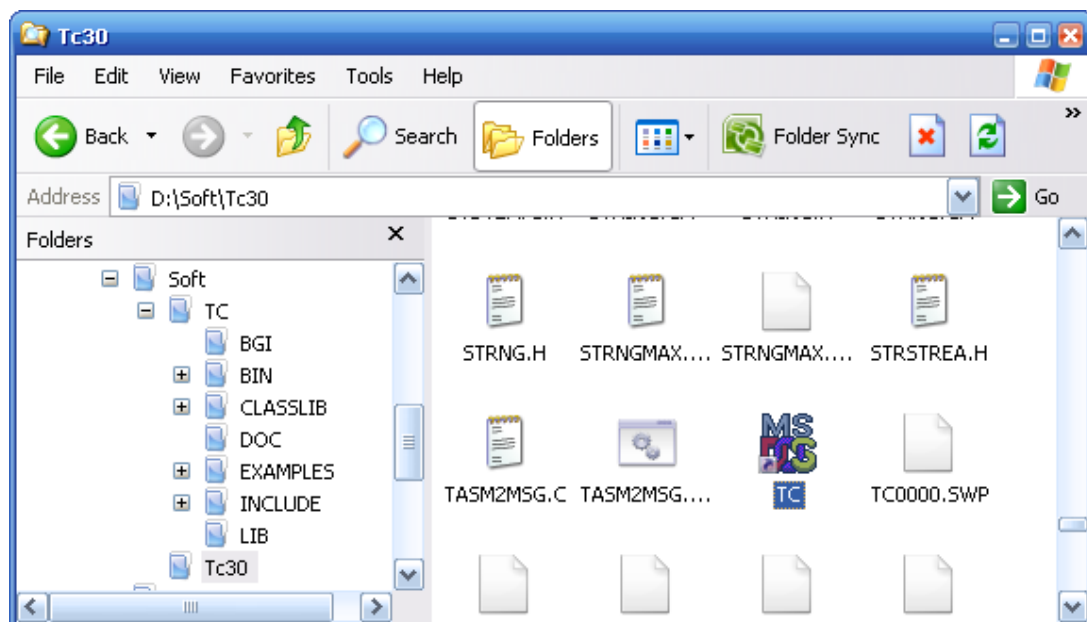
th c hành l p trình C, có th s d ng m t trong các ph n m m sau: TC ho c TC30 ho c BorlandC. phòng máy khu B tr ng C CN, sinh viên (SV) có th l a ch n TC ho c TC30.

Thông th ng, TC ho c TC30 c ch a trong a D (D:\Soft); n u không thì sinh viên có th tìm ki m b ng công c Search c a Window v i t khóa tìm ki m là TC (ho c TC30).

- N u s d ng TC thì SV m TC a D (D:\Soft\TC) → vào th m c BIN → double click chu t vào bi u t ng vàng có ch MSDS c tô m ch y TC nh hình sau:

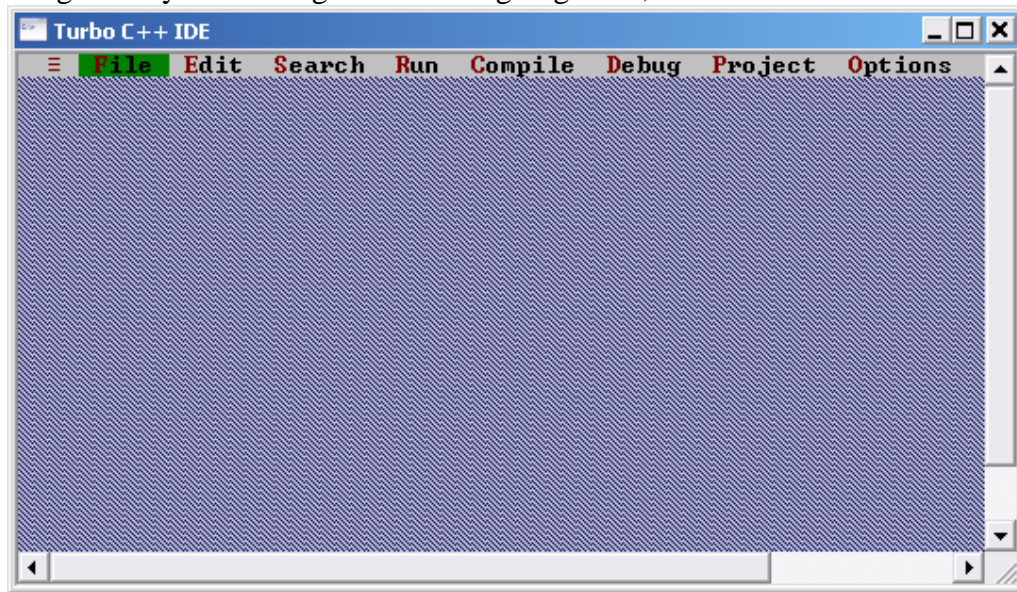


- Còn n u s d ng TC30 thì m TC30 trong a D(D:\Soft\TC30) → double click c ng vào bi u t ng vàng có ch MSDS c tô m ch y TC30. L u ý r ng, v i TC30 không có th m c BIN nh TC.

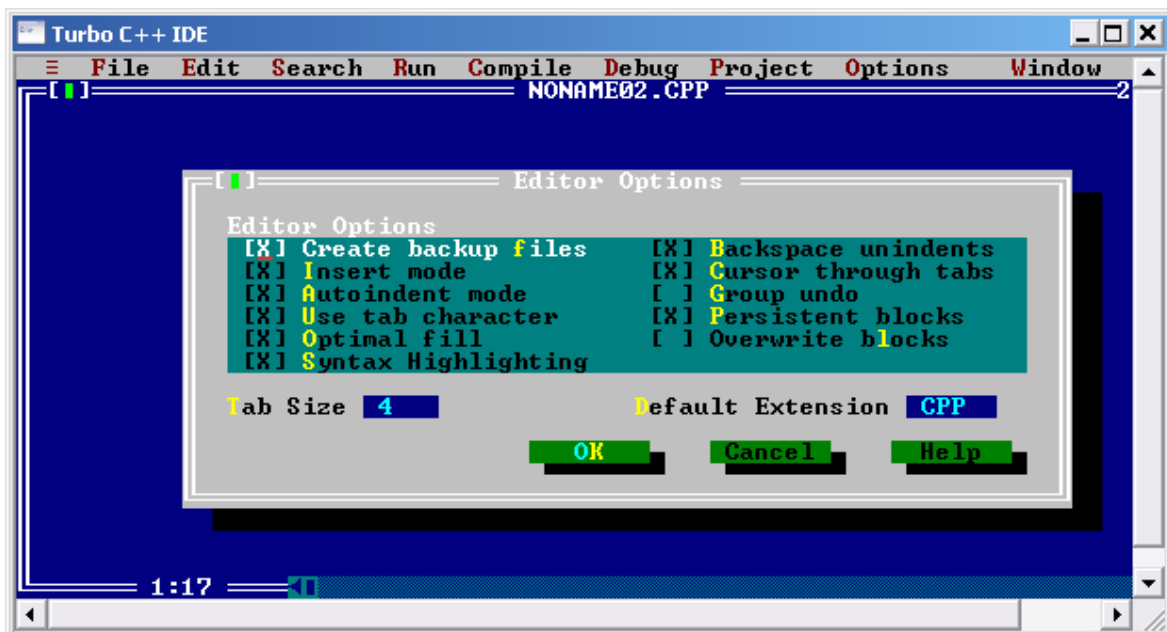


→ **L u ý:** sau khi m ch ng trình C xong, SV c n ph i nh ng d n ch a ch ng trình C ó ( sau này s l i ‘Unable to open...’). Nhìn vào 2 hình trên, ng d n ã m TC là **D:\Soft\TC\BIN**; còn ng d n ã m TC30 là **D:\Soft\TC30**.

- Dù s d ng TC hay TC30 thì giao di n v n gi ng nhau, nh sau:



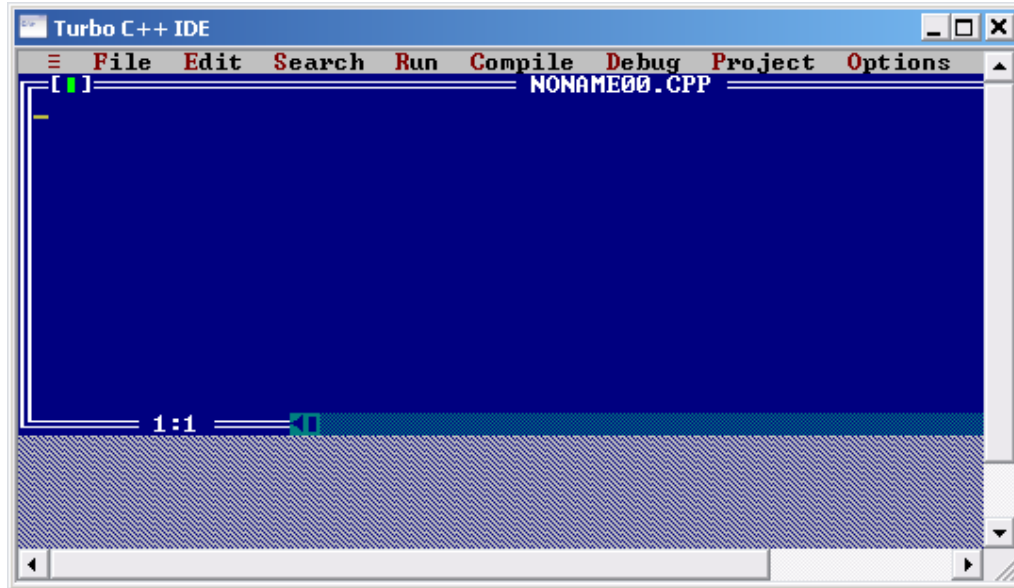
- Các phím th ng hay s d ng i v i ch ng trình C là:
  - + **Alt-Enter**: phóng to/thu nh màn hình C
  - + **F5**: phóng to c a s c a m t file ch ng trình \*.cpp ho c \*.c
  - + **F6**: chuy n i gi a các c a s file ch ng trình
  - + **F2**: l u m t file ch ng trình
  - + **F3**: m m t c a s m i t o m t file ch ng trình
  - + **Alt-F5**: xem k t qu ch y ch ng trình
  - + **F7/F8**: ch y ch ng trình t ng b c
  - + **F9**: biên d ch ch ng trình
  - + **Ctrl-F9**: ch y ch ng trình
- hi u ch nh trình bày trong C, vào Options → Environment → Editor, m t giao di n hi u ch nh hi n ra nh d i ây, s a i m c Tab size t 8 xu ng 4 kho ng cách Tab ng n h n.



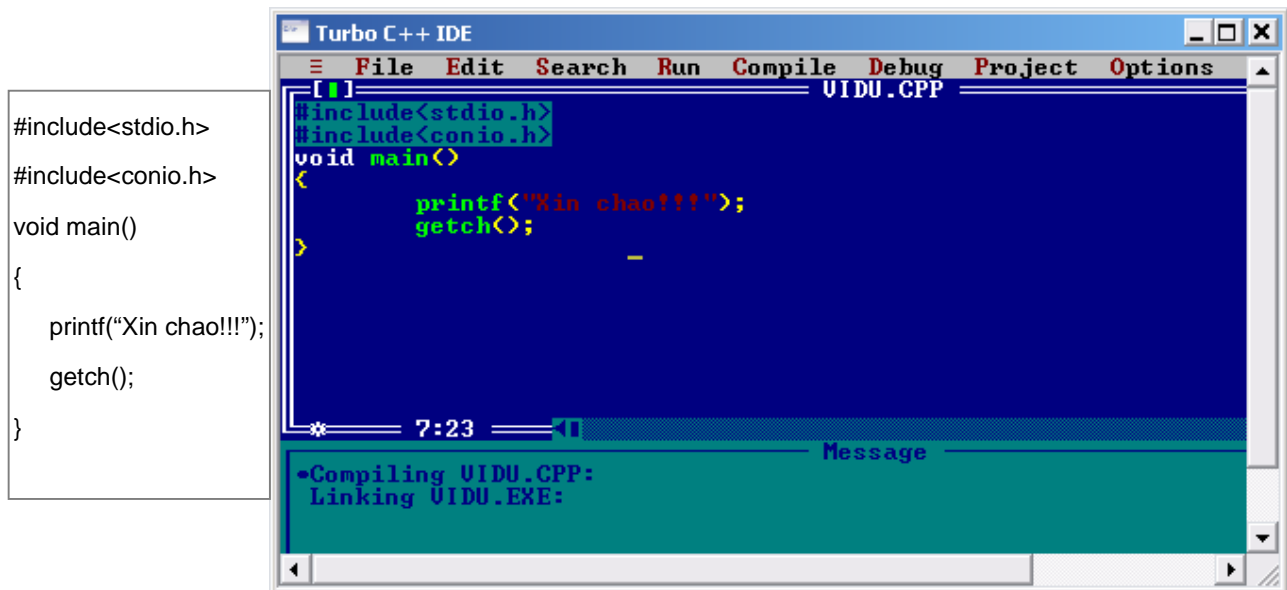
- **L u ý:** Khi trình bày m t ch ng trình C, SV nên bi t s d ng phím Tab trên bàn phím v i quy t c là t t c các câu l nh có cùng ph m vi nh h ng thì th ng hàng d c v i nhau, còn n u m t ho c nhi u câu l nh c th c hi n ph thu c vào m t câu l nh phía tr c nó thì canh l trái c a nó ph i c Tab vô m t l n Tab.

### 1) T o m t file ch ng trình trong C

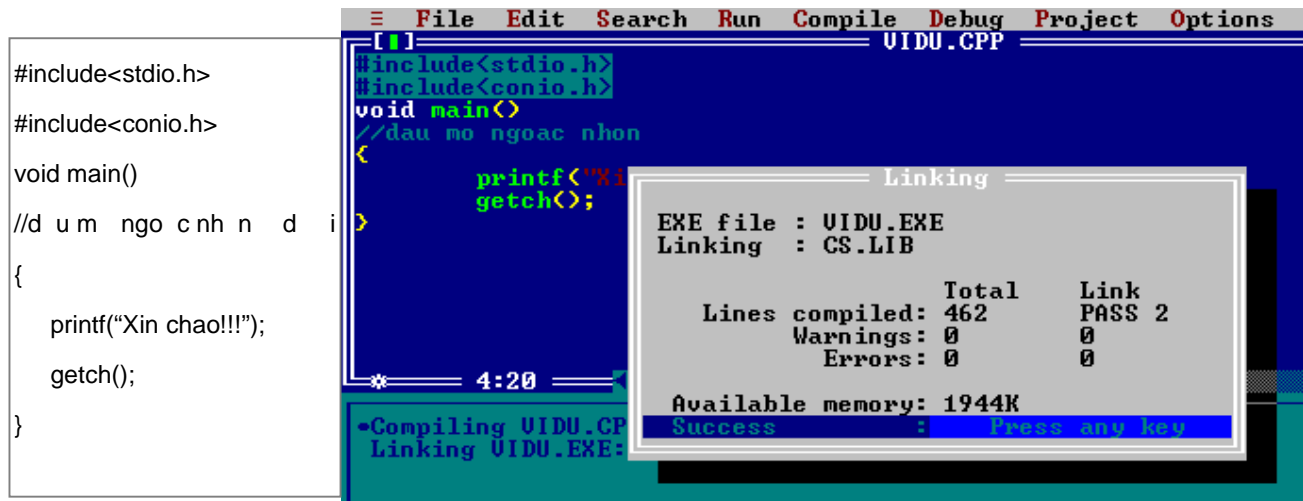
t o m t file ch ng trình trong C, sau khi ã m C xong → click vào menu File → ch n New (ho c c ng có th n phím F3), màn hình xu t hi n nh sau:



**Ví d :** t o m t ch ng trình l u v i tên Vidu.cpp xu t ra câu Xin chào!!!



Sau khi biên d ch ( n F9), n u xu t hi n câu thông báo ‘**Error ...**’ thì s d ng m c 2 s a l i; n u là ‘**Warning ...**’ thì ch mang tính c nh báo và không nh t thì t ph i s a i file ch ng trình (xem m c 3); còn n u thông báo có d ng ‘**Success : Press any key**’ nh d i ây thì có ngh a là ch ng trình vi t úng cú pháp và có th ch y c:



- Nhấn Enter để chạy chương trình, và sau đó nhấn Ctrl-F9 để chạy chương trình này, màn hình màu đen sẽ xuất hiện và hiển thị ra câu thông báo 'Xin chào!!!' theo yêu cầu:



- Nhấn Enter để giao diện xanh của chương trình.

Trong ví dụ trên, dòng đầu tiên `#include<stdio.h>` là khai báo sử dụng thư viện, mà chức năng trong chương trình có sử dụng câu lệnh `printf("Xin chào!!!")` thuộc thư viện `stdio.h`, do đó cần phải khai báo thư viện này.

Ngoài ra, nếu muốn biết thư viện `stdio.h` có chứa những câu lệnh nào, chỉ cần bôi đen một thư viện rồi nhấn Ctrl-F1 (đây là cách sử dụng Help – Trợ giúp trong phần mềm C++ như thường lệ), màn hình sẽ xuất hiện và liệt kê danh sách các câu lệnh thuộc thư viện này.



tr v l i màn hình ch ng trình, có th n phím F6 cho n khi xu t h i n c a s c a ch ng trình, ho c click vào d u ch m vuông xanh lá cây góc trên trái c a màn hình C theo ng m i tên trên.

T ng t , do trong ch ng trình có s d ng câu l nh getch()- d ng màn hình en khi ch y ch ng trình, mà l nh này thu c th v i n conio.h (ho c c ng có th b o i en c m t getch r i n Ctrl-F1 thì s b i t c câu l nh getch() thu c th v i n nào), nên c n ph i khai báo thêm th vi n này dòng th hai c a ch ng trình: #include<conio.h>.

### L u ý:

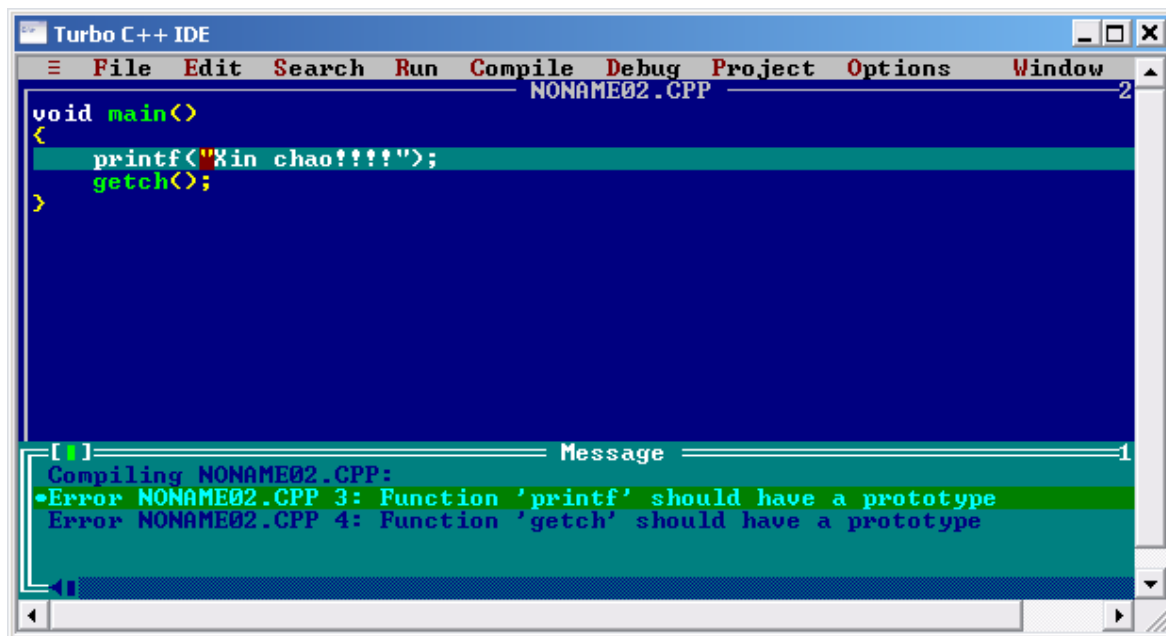
- B c c c a m t ch ng trình C luôn bao g m 3 ph n: ph n khai báo th v i n n m trên cùng c a ch ng trình; ph n hàm main() là hàm chính c a ch ng trình; và ký hi u b t u (d u '{' - begin) và k t thúc (d u '}' - end).
- Trong C luôn s d ng các hàm và m i hàm u ph i có giá tr tr v , và c hàm main() c ng v y. Chính vì v y, trong ví d trên có dòng l nh: **void main()** – có ngh a là hàm main() này không có giá tr tr v (không dùng câu l nh return)

## 2) Các l i và cách s a l i

D i ây là các l i thông d ng hay g p ph i khi s d ng ph n m m C, và v trí sai c phát h i n chính là dòng c tô m màu xanh:

### a) L i không khai báo th v i n cho các câu l nh c s d ng trong file ch ng trình

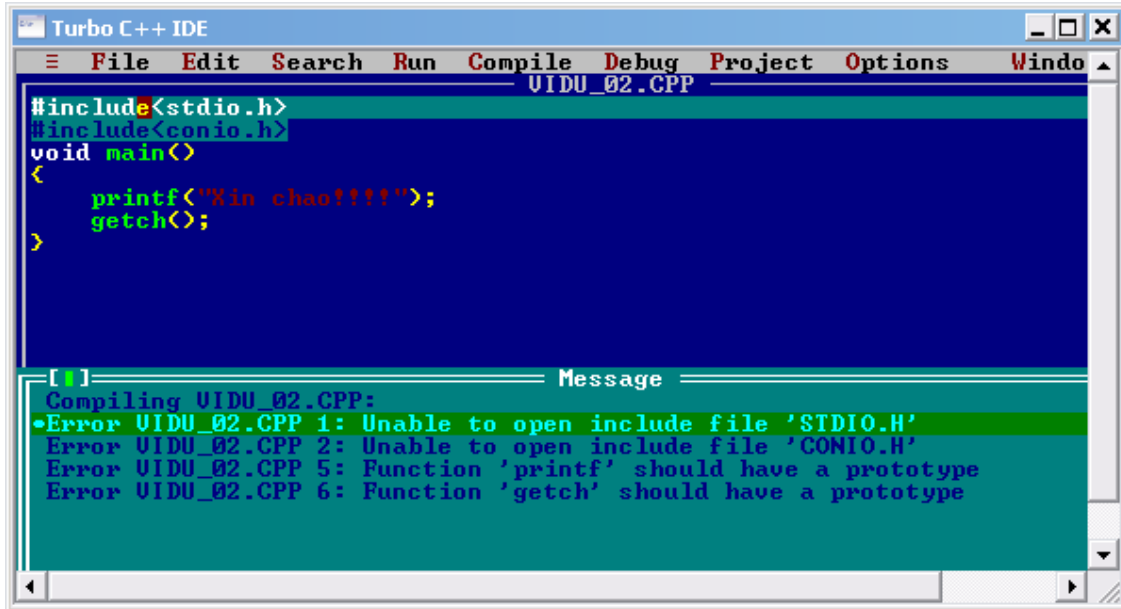
Câu thông báo cho l i này th ng g p là **Function 'tên câu l nh' should have a prototype** nh hình v d i, có ngh a là câu l nh 'tên câu l nh' c s d ng trong ch ng trình nh ng ch a khai báo th v i n ch a câu l nh này. (ho c gõ sai tên câu l nh)



- ➔ S a l i b ng cách b o i en 'tên câu l nh' trong file ch ng trình ➔ n Ctrl-F1 ➔ xác nh th v i n ch a câu l nh ➔ quay tr v c a s ch ng trình ➔ thêm vào file ch ng trình dòng khai báo th v i n t ng ng ➔ sau ó n F2 l u l i file ch ng trình v a c s a i và n F9 biên d ch tr l i.
- ➔ L p l i cho n khi không còn g p câu báo l i nh trên.

## b) L i không ch úng ng d n n các t p th vi n c a ch ng trình C

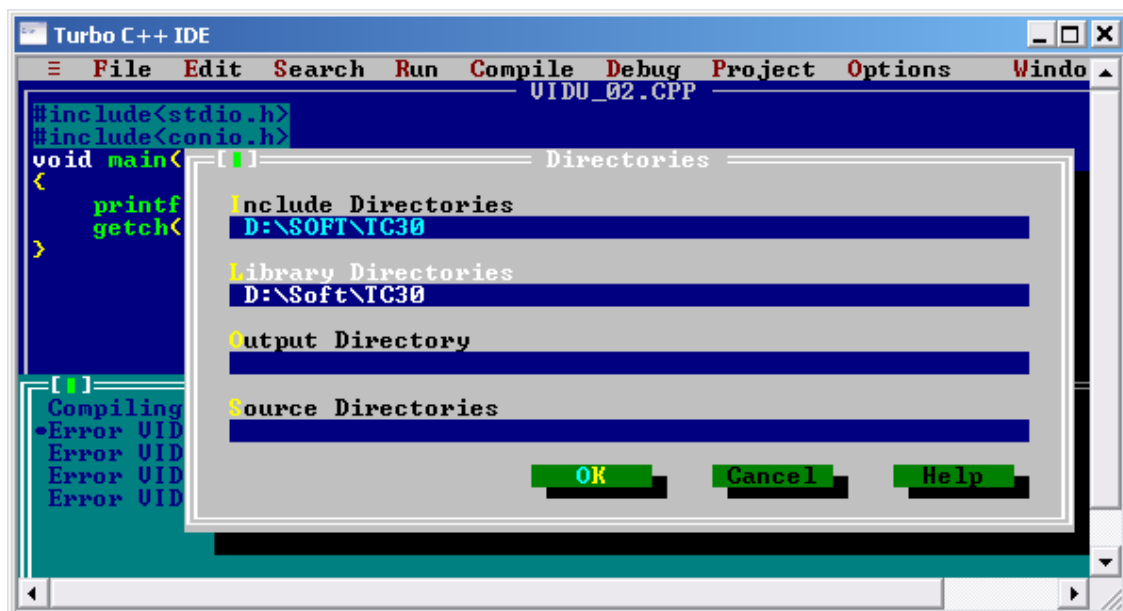
Câu thông báo l i th ng g p là Unable to open include file 'STDIO.H'/'CONIO.H' nh hình v d i, ngh a là ph n m m C không tìm th y các file th vi n mà ch ng trình có s d ng.



→ S a l i này b ng cách vào Options → Directories → gõ vào ng d n ch a file thu c th m c Include ô tr ng u tiên, và gõ vào ng d n ch a file thu c th m c Lib ô tr ng th hai → sau ó n OK.

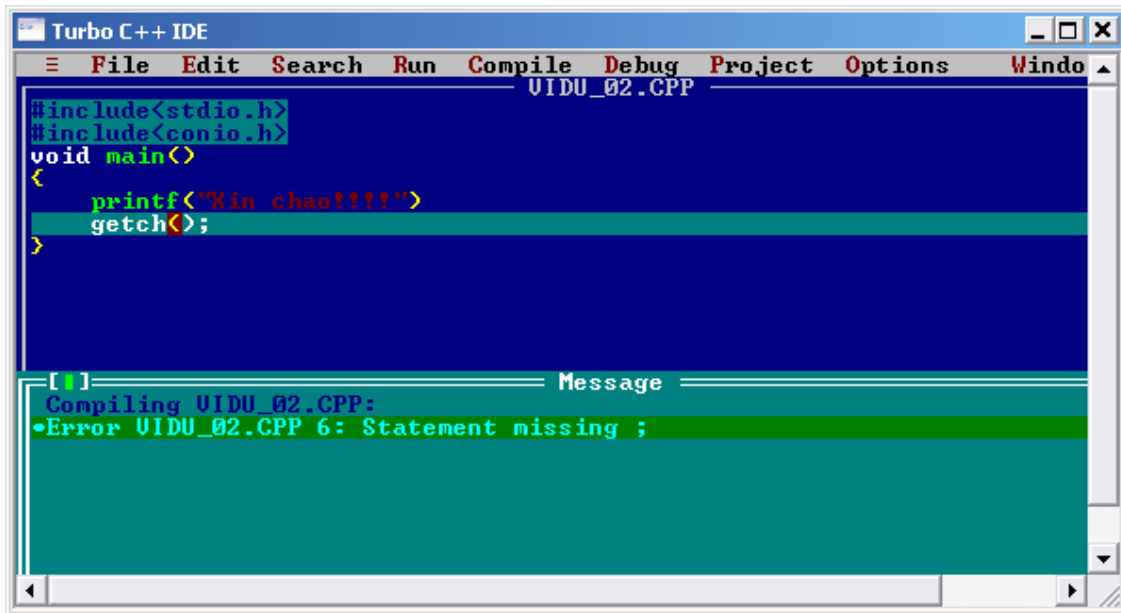
→ Có s khác bi t rõ ràng khi s d ng ph n m m TC và ph n m m TC30:

- + N u s d ng ph n m m TC thì ng gõ vào ô tr ng th nh t ng d n ch n th m c Include trong TC (c th ây là D:\Soft\TC\Include); và gõ vào ô tr ng th hai ng d n ch n th m c Lib trong TC (c th là D:\Soft\TC\Lib).
- + N u s d ng TC30 thì c 2 dòng u gõ vào ng d n ch a TC30 (c th là D:\Soft\TC30), b i vì trong TC30 không có ch a các th m c Include và Lib nh trong TC, ng th i các file th vi n trong TC30 u thu c tr c ti p th m c TC30 này.



### c) L i thi u d u ch m ph y (d u k t th u c m i c u u l n h t r o n g C)

C u u th o n g b a o l i t h n g g p l à **Statement missing** ; n h h n h d i , c ó n g h a l à t h i u m t d u c h m ph y m t d o n g l n h n à o ó t r o n g c h n g t r ì n h .



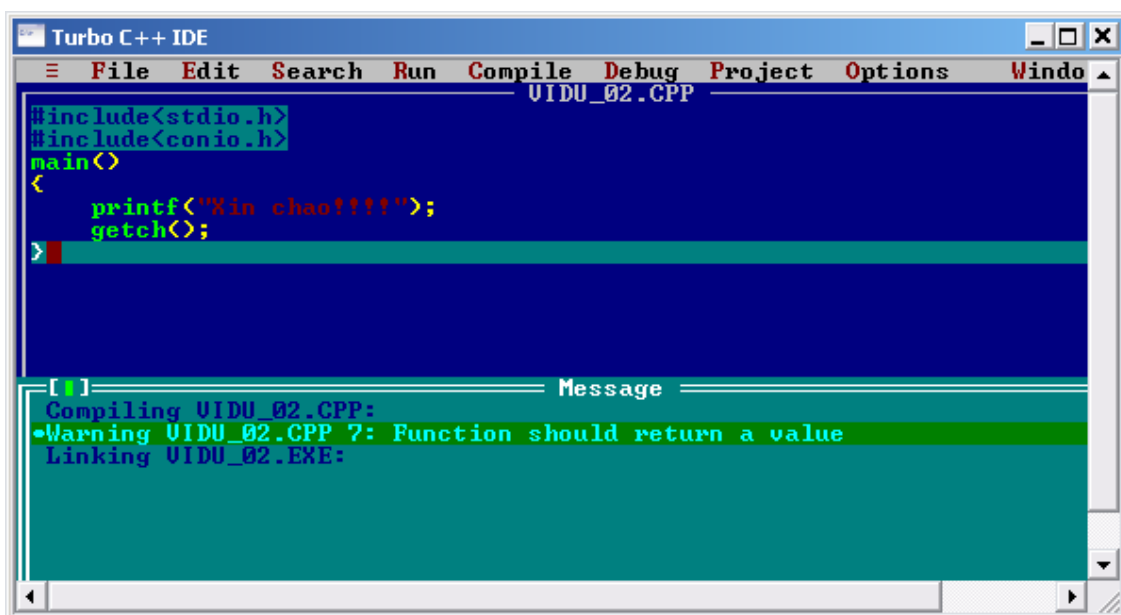
→ S a l i b n g c á c h x á c n h d o n g c t ô m m à u x a n h , v à d u c h m ph y t h o n g t h n g s b t h i u n g a y d o n g k t r ê n d o n g t ô x a n h m n à y (h o c ô i k h i n g a y c h í n h d o n g ó h o c d o n g k s a u n ó).

### 3) Các c nh báo (Warning)

C á c c n h b á o t h n g k h o n g n h h n g n v i c c h y c h n g t r ì n h , t u y n h i ê n k h i c ó c n h b á o t h i c h ú n g t a c n g n ê n i u c h n h c á c d o n g l n h k h c p h c c n h b á o ó .

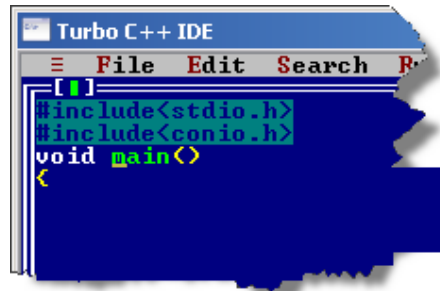
#### a) C nh báo ‘Function should return a value’

C n h b á o n à y c h o b i t r n g t r o n g c h n g t r ì n h c ó s d n g h à m (v í d h à m m a i n ( ) ) n h n g k h o n g k h a i b á o k i u g í á t r t r v c a h à m n h h n h n h s a u :



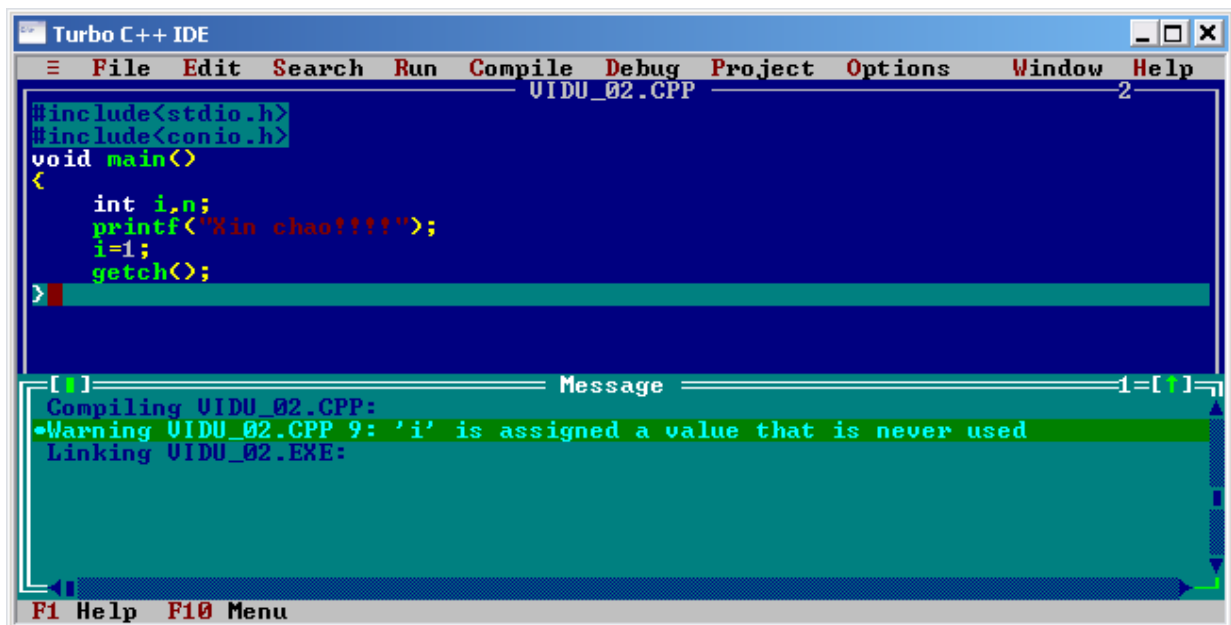


- Kh c ph c c nh báo này b ng cách thêm vào phía tr c ch main() t khóa void nh d i này:



#### b) C nh báo ‘... i assigned a value that is never used’

C nh báo này cho bi t trong ch ng trình có khai báo bi n, gán giá tr cho bi n nh ng l i không s d ng bi n này → kh c ph c c nh báo này b ng cách xóa i bi n c khai ó.



#### 4) Ch ng trình ch y vô h n

L i này x y ra khi ã biên d ch ch ng trình ( n F9) thành công và sau khi n Ctrl-F9 ch y ch ng trình thì nó ch y vô h n không th thoát ra tr l i c a s ch ng trình c sau khi n Enter nhi u l n.

- t m d ng ch ng trình và tr v c a s ch ng trình, gõ t h p phím Ctrl-Break → sau ó gõ phím Ctrl-F2 làm m t dòng c tô màu xanh m.

- T i n hành ch nh s a code ch ng trình và l i l u, biên d ch và ch y ch ng trình.

#### 5) Không xem c k t qu ch y ch ng trình

L i này x y ra sau khi n t h p phím Ctrl-F9 mà v n không xem c k t qu ch y ch ng trình, ng th i v n gi nguyên c a s file ch ng trình.

- Kh c ph c b ng cách n t h p phím Alt-F5 xem k t qu ch y ch ng trình ho c vào ch nh s a code (thêm dòng getch()) k cu i ch ng trình).

#### 6) Các nh d ng hay s d ng i v i các ki u d li u

+ Ki u d li u int: nh d ng %d

- + Ki u d li u float: nh d ng %f
- + Ki u d li u long int: nh d ng %ld
- + Ki u d li u double: nh d ng %lf
- + Ki u d li u char: nh d ng %c
- + Ki u d li u chu i (khai báo char[chi u dài t i a]): nh d ng %s

## 7) Khai báo và s d ng chu i ký t trong C

Trong C không có ki u d li u chu i ký t (xâu). Chính vì v y, s d ng c chu i ký t (t c là m t bi n nh n giá tr ch a nhi u h n m t ký t , ví d nh bi n hoten th ng nh n giá tr là ‘Nguyen Van A’ ho c ‘Le Thi B’) thì c n ph i khai báo bi n ó có ki u d li u char và dài c a nó.

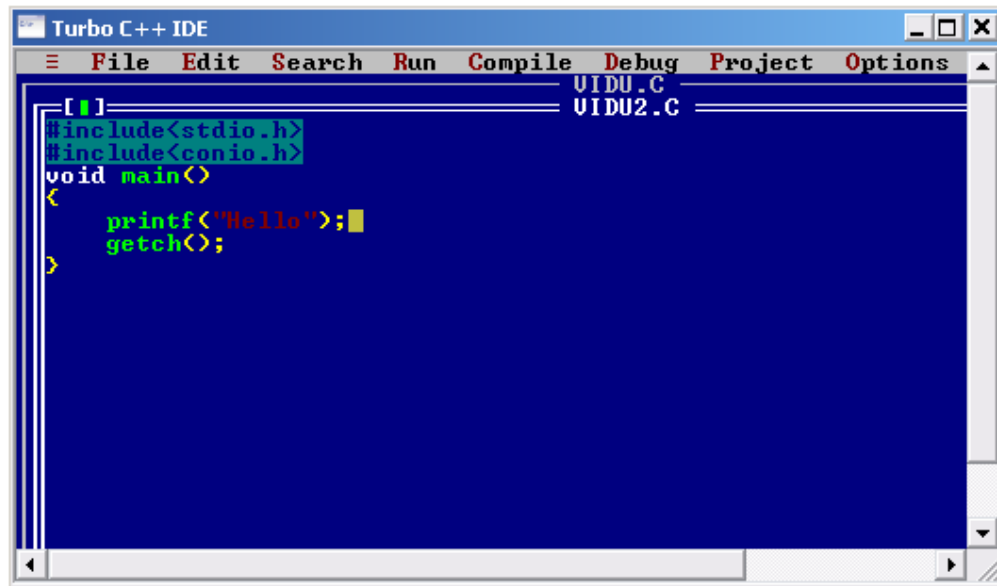
**Ví d :** gi s ch ng trình yêu c u khai m t bi n hoten nh n giá tr c a m t chu i ký t nh p vào nh ‘Nguyen Thi Ha Quyen’ thì c n khai báo nh sau:

**char hoten[30];**

Con s 30 cho bi t bi n h tên này ch nh n chu i ký t có chi u dài không nhi u h n 30 ký t .

## 8) Chuy n ch vi t è sang ch chèn

Con tr nh p nháy trong C thông th ng có d ng là m t d u g ch ngang. Tuy nhiên, n u vô tình chúng ta n vào phím Insert trên bàn phím thì con nháy s bi n thành d ng hình ch nh t tô màu vàng, và lúc này chúng ta không th n Enter xu ng dòng c ng nh ch so n th o trong C chuy n sang ch vi t è.



→ Kh c ph c l i này b ng cách n phím Insert m t l n n a, con tr nh p nháy s tr v d ng d u g ch ngang nh bình th ng.

## 9) Các câu l nh n trong C th ng dùng

M t câu l nh (statement) xác nh m t công vi c mà ch ng trình ph i th c hi n x lý d li u ã c mô t và khai báo. Các câu l nh c ng n cách v i nhau b i d u ch m ph y (;).

**Phân lo i:** có hai lo i l nh, ó là l nh n và l nh có c u trúc.

- L nh n là m t l nh không ch a các l nh khác. Các l nh n g m: l nh gán, các câu l nh nh p xu t d li u...
- L nh có c u trúc là l nh trong ó ch a các l nh khác. L nh có c u trúc bao g m: c u trúc i u ki n r nhánh, c u trúc i u ki n l a ch n, c u trúc l p và c u trúc l nh h p thành. L nh h p thành (kh i l nh) là m t nhóm bao g m nhi u khai báo bi n và các l nh c gom vào trong c p d u { }.

### a) L nh gán

L nh gán (assignment statement) dùng gán giá tr c a m t bi u th c cho m t bi n.

- Cú pháp: **<Tên bi n> = <bi u th c>**
- Ví d :

```
int main()
{
    int x,y;
    x=10; /*Gán h ng s 10 cho bi n x*/
    y = 2*x; /*Gán giá tr 2*x=2*10=20 cho x*/
    return 0;
}
```

Nguyên t c khi dùng l nh gán là ki u c a bi n và ki u c a bi u th c ph i gi ng nhau, g i là có s t ng thích gi a các ki u d li u. Ch ng h n ví d sau cho th y m t s không t ng thích v ki u:

```
int main()
{
    int x,y;
    x = 10; /*Gán h ng s 10 cho bi n x*/
    y = "Xin chao";
    /*y có ki u int, còn "Xin chao" có ki u char* */
    return 0;
}
```

Khi biên d ch ch ng trình này, C s báo l i "Cannot convert 'char\*' to 'int'" t c là C không th t ng chuy n i ki u t char\* (chu i ký t ) sang int.

Tuy nhiên trong a s tr ng h p s t ng bi n i ki u s t ng thích v ki u s c th c hi n.

Ví d :

```
int main()
{
    int x,y;
    float r;
    char ch;
    r = 9000;
```

```

x = 10; /* Gán h ng s 10 cho bi n x */
y = 'd'; /* y có ki u int, còn 'd' có ki u char*/
r = 'e'; /* r có ki u float, 'e' có ki u char*/
ch = 65.7; /* ch có ki u char, còn 65.7 có ki u float*/
return 0;
}

```

Trong nhi u tr ng h p t o r a s t ng thích v ki u, ta ph i s d ng n cách th c chuy n i ki u m t cách t ng minh.

Cú pháp c a phép toán này nh sau: **(Tên ki u) <Bi u th c>**

Chuy n i ki u c a <Bi u th c> thành ki u m i <Tên ki u>. Ch ng h n nh :

```

float f;
f = (float) 10 / 4; /* f lúc này là 2.5*/

```

Chú ý:

- Khi m t bi u th c c gán cho m t bi n thì giá tr c a nó s thay th giá tr c mà bi n ã l u gi tr c ó.
- Trong câu l nh gán, d u = là m t toán t ; do ó nó có th c s d ng là m t thành ph n c a bi u th c. Trong tr ng h p này giá tr c a bi u th c gán chính là giá tr c a bi n.

Ví d :

```

int x, y;
y = x = 3; /* y lúc này cùng b ng 3*/

```

Ta có th gán tr cho bi n lúc bi n c khai báo theo cách th c sau:

**<Tên ki u> <Tên bi n> = <Bi u th c>;**

Ví d : int x = 10, y=x;

### **b) L nh nh p giá tr t bàn phím cho bi n (hàm scanf)**

Hàm scanf là hàm cho phép c d li u t bàn phím và gán cho các bi n trong ch ng trình khi ch ng trình th c thi. Trong ngôn ng C, ó là hàm scanf n m trong th vi n stdio.h.

Cú pháp: **scanf("Chu i nh d ng", a ch c a các bi n);**

Gi i thích:

- Chu i nh d ng: dùng qui nh ki u d li u, cách bi u di n, r ng, s ch s th p phân... M t s nh d ng khi nh p ki u s nguyên, s th c, ký t .

nh d ng	Ý ngh a
%[s ký s ]d	Nh p s nguyên có t i a <s ký s >
%[s ký s ]f	Nh p s th c có t i a <s ký s > tính c d u ch m
%c	Nh p m t ký t

Ví d :

```

%d      Nh p s nguyên
%4d     Nh p s nguyên t i a 4 ký s , n u nh p nhi u h n 4 ký s thì ch nh n c 4
        ký s u tiên
%f      Nh p s th c

```

%6f      Nh p s th c t i a 6 ký s (tính luôn d u ch m), n u nh p nhi u h n 6 ký s thì ch nh n c 6 ký s u tiên (ho c 5 ký s v i d u ch m)

- a ch c a các bi n: là a ch (&) c a các bi n mà chúng ta c n nh p giá tr cho nó. c vì t nh sau: &<tên bi n>.

Ví d :

```
scanf("%d",&bien1);/*Doc gia tri cho bien1 co kieu nguyen*/
scanf("%f",&bien2);/*Doc gia tri cho bien2 co kieu th c*/
scanf("%d%f",&bien1,&bien2);
/*Doc gia tri cho bien1 co kieu nguyen, bien2 co kieu thuc*/
scanf("%d%f%c",&bien1,&bien2,&bien3);
/*bien3 co kieu char*/
```

L u ý:

- + Chu i nh d ng ph i t trong c p d u nháy kép ("").
- + Các bi n ( a ch bi n) ph i cách nhau b i d u ph y (,).
- + Có bao nhiêu bi n thì ph i có b y nhiêu nh d ng.
- + Th t c a các nh d ng ph i phù h p v i th t c a các bi n.
- + nh p giá tr ki u char c chính xác, nên dùng hàm fflush(stdin) lo i b các ký t còn n m trong vùng m bàn phím tr c hàm scanf().
- + nh p vào m t chu i ký t (không ch a kho ng tr ng hay k t thúc b ng kho ng tr ng), chúng ta ph i khai báo ki u m ng ký t hay con tr ký t , s d ng nh d ng %s và tên bi n thay cho a ch bi n.
- + c vào m t chu i ký t có ch a kho ng tr ng (k t thúc b ng phím Enter) thì ph i dùng hàm gets().

Ví d :

```
int biennguyen;
float bienthuc;
char bienchar;
char chuoi1[20], *chuoi2;
```

Nh p giá tr cho các bi n:

```
scanf("%3d",&biennguyen);
```

- N u ta nh p 1234455 thì giá tr c a biennguyen là 3 ký s u tiên (123). Các ký s còn l i s còn n m l i trong vùng m.

```
scanf("%5f",&bienthuc);
```

- N u ta nh p 123.446 thì giá tr c a bienthuc là 123.4, các ký s còn l i s còn n m trong vùng m.

```
scanf("%2d%5f",&biennguyen, &bienthuc);
```

- N u ta nh p liên ti p 2 s cách nhau b i kho ng tr ng nh sau: 1223 3.142325, thì:

2 ký s u tiên (12) s c c vào cho biennguyen.

2 ký s t i p theo tr c kho ng tr ng (23) s c c vào cho bienthuc.

scanf("%2d%5f%c",&biennguyen, &bienthuc,&bienchar)

- N u ta nh p liên ti p 2 s cách nhau b i kho ng tr ng nh sau: 12345 3.142325:

2 ký s u tiên (12) s c c vào cho biennguyen.

3 ký s t i p theo tr c kho ng tr ng (345) s c c vào cho bienthuc.

Kho ng tr ng s c c cho bienchar.

- N u ta ch nh p l s g m nhi u ký s nh sau: 123456789:

2 ký s u tiên (12) s c c vào cho biennguyen.

5 ký s t i p theo (34567) s c c vào cho bienthuc.

bienchar s có giá tr là ký s t i p theo '8'.

scanf("%s",chuoi1); ho c scanf("%s",chuoi2)

- N u ta nh p chu i nh sau: Nguyen Van Linh (↵) thì giá tr c a bi n chuoi1 hay chuoi2 ch là Nguyen .

scanf("%s%s",chuoi1, chuoi2);

- N u ta nh p chu i nh sau: Duong Van Hieu (↵) thì giá tr c a bi n chuoi1 là Duong và giá tr c a bi n chuoi2 là Van.

- Vì sao nh v y? C s c t u n khi g p kho ng tr ng và gán giá tr cho bi n u tiên, ph n còn l i sau kho ng tr ng là giá tr c a các bi n t i p theo.

gets(chuoi1);

- N u nh p chu i : Nguyen Van Linh (↵) thì giá tr c a bi n chuoi1 là Nguyen Van Linh

### c) L nh xu t giá tr c a bi u th c lên màn hình (hàm printf)

Hàm printf (n m trong th vi n stdio.h) dùng xu t giá tr c a các bi u th c lên màn hình.

**Cú pháp:** printf("Chu i nh d ng", Các bi u th c);

**Gì i thích:**

- Chu i nh d ng: dùng qui nh ki u d li u, cách bi u di n, r ng, s ch s th p phân... M t s nh d ng khi i v i s nguyên, s th c, ký t .

nh d ng	Ý ngh a
%d	Xu t s nguyên
%[.s ch s th p phân]f	Xu t s th c có <s ch s th p phân> theo quy t c làm tròn s .
%o	Xu t s nguyên h bát phân
%x	Xu t s nguyên h th p l c phân
%c	Xu t m t ký t
%s	Xu t chu i ký t
%e ho c %E ho c %g ho c %G	Xu t s nguyên d ng khoa h c (nhân 10 m x)

**Ví d :**

%d In ra s nguyên

%4d In s nguyên t i a 4 ký s , n u s c n in nhi u h n 4 ký s thì in h t

%f In s th c

%6f In s th c t i a 6 ký s (tính luôn d u ch m), n u s c n in nhi u h n 6 ký s thì in h t

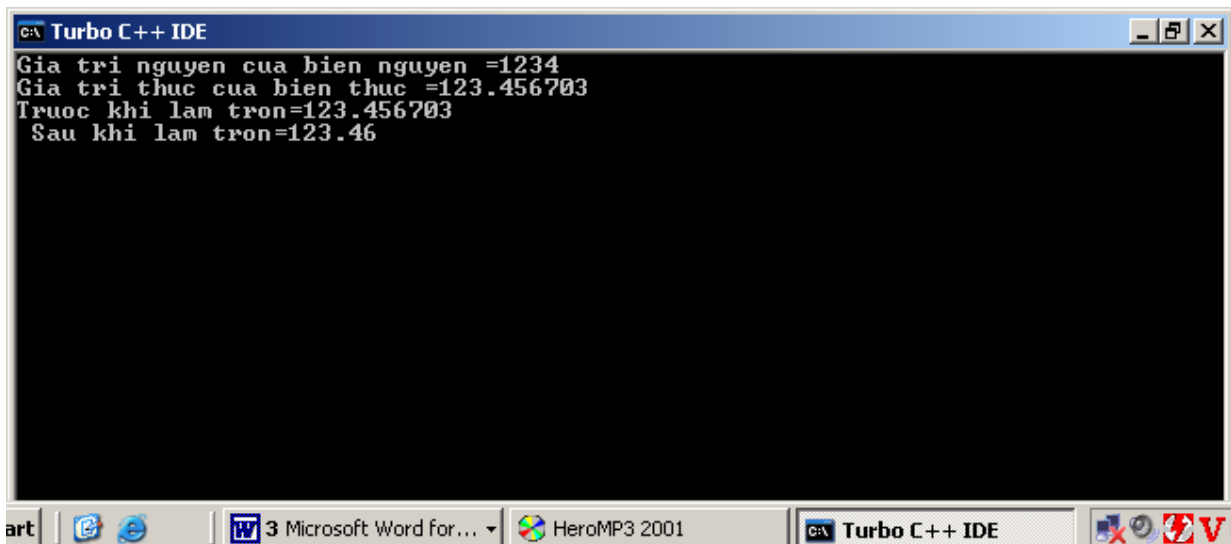
%.3f In s th c có 3 s l , n u s c n in có nhi u h n 3 s l thì làm tròn.

- Các bi u th c: là các bi u th c mà chúng ta c n xu t giá tr c a nó lên màn hình, m i bi u th c phân cách nhau b i d u ph y (.).

Ví d :

```
include<stdio.h>
int main()
{
    int bien_nguyen=1234, i=65;
    float bien_thuc=123.456703;
    printf("Gia tri nguyen cua bien nguyen =%d\n",bien_nguyen);
    printf("Gia tri thuc cua bien thuc =%f\n",bien_thuc);
    printf("Truoc khi lam tron=%f \n
    Sau khi lam tron=%.2f",bien_thuc, bien_thuc);
    return 0;
}
```

K t qu in ra màn hình nh sau:



L u ý:

iv i các ký t i u khi n, ta không th s d ng cách vi t thông th ng hi n th chúng.

Ký t i u khi n là các ký t dùng i u khi n các thao tác xu t, nh p d li u.

M t s ký t i u khi n c mô t trong b ng:

Ký t i u khi n	Giá tr th p l c phân	Ký t c hi n th	Ý ngh a
\a	0x07	BEL	Phát ra ti ng chuông
\b	0x08	BS	Di chuy n con tr sang trái 1 ký t và xóa ký t bên trái (backspace)
\f	0x0C	FF	Sang trang
\n	0x0A	LF	Xu ng dòng
\r	0x0D	CR	Tr v u dòng

\t	0x09	HT	Tab theo c t (gi ng gõ phím Tab)
\\	0x5C	\	D u \
\'	0x2C	'	D u nháy n (')
\"	0x22	"	D u nháy kép (")
\?	0x3F	?	u ch m h i (?)
\ddd	ddd	Ký t có mã ACSII trong h bát phân là s ddd	
\xHHH	oxHHH	Ký t có mã ACSII trong h th p l c phân là HHH	

Ví d :

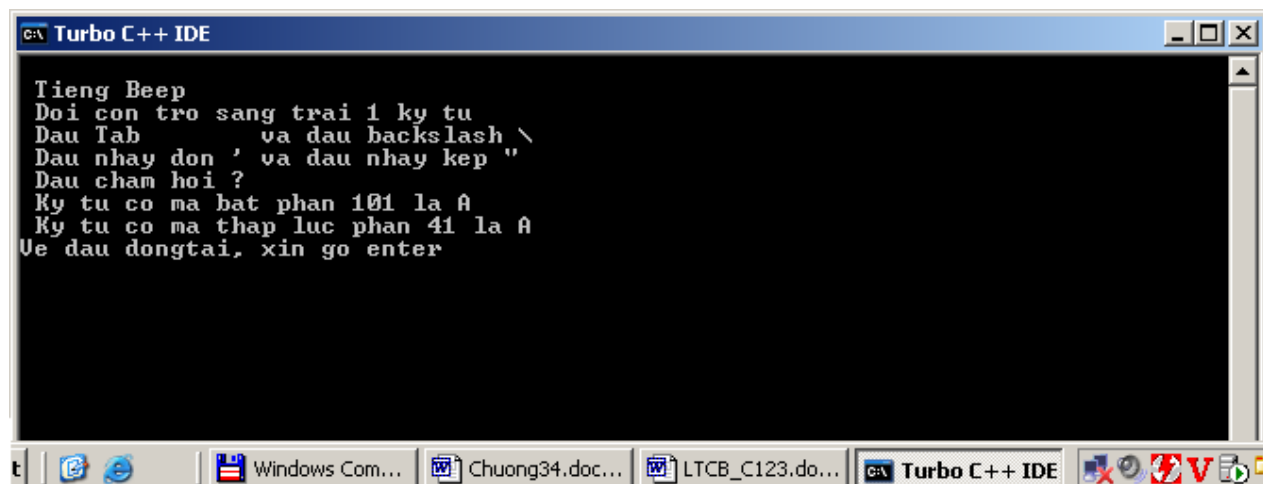
```
#include <stdio.h>
#include<conio.h>
int main ()
{
    clrscr();
    printf("\n Tieng Beep \a");
    printf("\n Doi con tro sang trai 1 ky tu\b");
    printf("\n Dau Tab \tva dau backslash \\");
    printf("\n Dau nhay don \' va dau nhay kep '\"");
    printf("\n Dau cham hoi \?");
    printf("\n Ky tu co ma bat phan 101 la \101");
    printf("\n Ky tu co ma thap luc phan 41 la \x041");
    printf("\n Dong hien tai, xin go enter");
    getch();
    printf("\rVe dau dong");
    getch();
    return 0;
}
```

K t qu tr c khi gõ phím Enter:

```
C:\ Turbo C++ IDE
Tieng Beep
Doi con tro sang trai 1 ky tu
Dau Tab va dau backslash \
Dau nhay don ' va dau nhay kep "
Dau cham hoi ?
Ky tu co ma bat phan 101 la A
Ky tu co ma thap luc phan 41 la A
Dong hien tai, xin go enter
```

K t qu sau khi gõ phím Enter:





The image shows a screenshot of the Turbo C++ IDE window. The title bar reads "C:\ Turbo C++ IDE". The main text area contains the following C code:

```
Tieng Beep
Doi con tro sang trai 1 ky tu
Dau Tab      va dau backslash \
Dau nhay don ' va dau nhay kep "
Dau cham hoi ?
Ky tu co ma bat phan 101 la A
Ky tu co ma thap luc phan 41 la A
Ve dau dongtai, xin go enter
```

The taskbar at the bottom shows several open applications: "Windows Com...", "Chuong34.doc...", "LTCB\_C123.do...", and "Turbo C++ IDE". There are also system icons on the right side of the taskbar.

# BÀI THI C HÀNH S 1

(Ch ơ ng I: CÁC THÀNH PHẦN C Ơ B Ơ N C Ơ A NGÔN NG Ữ L ẬP TRÌNH C)

Hãy c ơ th ơ t k ơ ph ơ n h ơ m và bi ơ u th ơ c trong gi ơ o tr ơ i n C, và làm các bài t ơ p sau:

- 1) Vi ơ t ch ơ ng tr ơ i n nh ơ p vào m ơ t s ơ ng uy ơ n n và x. Hãy cho bi ơ t th ơ ng uy ơ n n và s ơ d ơ c ơ a phép chia n cho x; và phép chia ph ơ n c ơ a n cho x.

**H ơ ng d ơ n:** s ơ d ơ ng phép toán s ơ h ơ c /(phép chia th ơ ng uy ơ n n), %(phép chia l ơ y s ơ d ơ ), và chuy ơ n i s ơ ng uy ơ n n ho ơ c x thành s ơ th ơ c.

- 2) Vi ơ t ch ơ ng tr ơ i n nh ơ p vào m ơ t s ơ th ơ c x và t ơ i n gi ơ a tr ơ y c ơ a các bi ơ u th ơ c sau:

a.  $y = x^7 \cdot e^x + 3x - 8$

b.  $y = (x^x + 9) / (5x + 7)$

c.  $y = |5x - 30| \cdot (2x^2 + 9)$

d.  $y = \sqrt{7x + 15}$

e.  $y = \sqrt[3]{25x^3 - 9}$  (c ơ n b ơ c 7)

**H ơ ng d ơ n:** s ơ d ơ ng các h ơ m chu ơ n trong C g ơ m h ơ m pow(double x, double y) – t ơ i n  $x^y$ , h ơ m exp(x) – t ơ i n  $e^x$ , h ơ m fabs(x) – t ơ i n tr ơ i n t ơ i c ơ a m ơ t s ơ th ơ c x và h ơ m sqrt(x) – t ơ i n c ơ n b ơ c 2 c ơ a x.

- 3) Vi ơ t ch ơ ng tr ơ i n nh ơ p vào các gi ơ a tr ơ i n tr ơ R1, R2, R3 c ơ a m ơ t m ơ ch ơ i n và t ơ i n t ơ ng tr ơ i n theo c ơ i n g th ơ c:

$$1/R = 1/R1 + 1/R2 + 1/R3$$

- 4) Vi ơ t ch ơ ng tr ơ i n nh ơ p vào h ơ t ơ n và tu ơ i c ơ a m ơ t sinh vi ơ n, r ơ i xu ơ t ra các th ơ i n g t ơ i n v ơ a nh ơ p.

**H ơ ng d ơ n:** do y ơ u c ơ u nh ơ p h ơ t ơ n và tu ơ i n Ơ n c ơ n khai b ơ o 2 bi Ơ n: hoten và tuoi. H ơ t ơ n c ơ a m ơ t ng Ơ i th Ơ i nh Ơ i h Ơ n m Ơ t k Ơ t Ơ cho Ơ n ph Ơ i khai b Ơ o nh Ơ sau: char hoten[30] (v Ơ i 30 l Ơ chi Ơ u d Ơ i t Ơ i a c Ơ phép nh Ơ p cho h Ơ t Ơ n m Ơ t ng Ơ i).

Ơ n ch Ơ ng tr Ơ i n c Ơ d Ơ ng sau:

```
char hoten[30];
int tuoi;
printf("Nhap ho ten : ");
gets(hoten); //c Ơ i l Ơ nh gets d Ơ ng l Ơ y các k Ơ t Ơ v Ơ a nh Ơ p và g Ơ n cho bi Ơ n hoten
fflush(stdin) //x Ơ a s Ơ ch b Ơ m m Ơ i khi nh Ơ p d Ơ l Ơ u gi Ơ a k Ơ u chu Ơ i và k Ơ u s
printf("Nhap tuoi : ");
scanf("%d",&tuoi);
printf("Sinh vien %s - %d tuoi",hoten,tuoi);
//v Ơ i bi Ơ n hoten l Ơ m Ơ t chu Ơ i k Ơ t Ơ cho Ơ n ph Ơ i s Ơ d Ơ ng nh Ơ d Ơ ng %s khi xu Ơ t ra m Ơ n h Ơ i n
```

- 5) Vi Ơ t m Ơ t Ơ n m Ơ gi Ơ và v Ơ m Ơ t l Ơ u nh Ơ p m Ơ t gi Ơ a tr Ơ l Ơ  $^{\circ}\text{C}$  (Celsius) và chuy Ơ n n Ơ sang  $^{\circ}\text{F}$  (Fahrenheit). [H Ơ ng d Ơ n:  $C/5 = (F-32)/9$ ]

## BÀI THI C HÀNH S 2

(Chương II: CÁC L NH C A NGÔN NG L P TRÌNH C)

- 1) Viết chương trình nhập vào một số thực x từ bàn phím. Hãy xuất ra màn hình câu thông báo “Bạn nhập đúng yêu cầu” nếu  $x \in [5,10]$ ; nếu không thì xuất ra câu thông báo “Bạn nhập x nằm ngoài đoạn  $[5,10]$ ”.

**Hướng dẫn:** sử dụng biểu thức logic (&& hoặc ||).

```
float x;
printf("%s", (x <= 5 && x >= 10) ? "Ban da nhap dung yeu cau": "Ban da nhap x nam ngoai doan [5,10]");
```

- 2) Viết chương trình nhập vào 4 số thực từ bàn phím. Hãy xuất ra màn hình giá trị lớn nhất và giá trị nhỏ nhất trong 4 số vừa nhập.

**Hướng dẫn:** sử dụng biểu thức logic

```
float x,y,u,v;
float min,max;
max=((x>=y?x:y) >= (u>=v?u:v)) ? (x>=y?x:y) : (u>=v?u:v);
min=((x<=y?x:y) <= (u<=v?u:v)) ? (x<=y?x:y) : (u<=v?u:v);
```

Hoặc có thể khai báo thêm các biến min\_xy, max\_xy, min\_uv và max\_uv cho giá trị min và max của tập x-y, u-v.

- 3) Viết chương trình nhập vào chiều dài hai đáy a, b và chiều cao h của hình thang. Hãy tính và xuất ra màn hình diện tích của hình thang có 3 chiều dài vừa nhập.

**Hướng dẫn:** sử dụng công thức tính diện tích hình thang  $dt = (a+b)*h/2$ ;

- 4) Viết chương trình nhập vào 3 hệ số của phương trình có dạng  $ax^2+bx+c=0$ . Hãy biện luận và xuất ra các nghiệm của phương trình này.

**Hướng dẫn:** biện luận từng trường hợp để phân loại phương trình bậc hai suy biến thành phương trình bậc nhất thay bậc 0 phụ thuộc vào các hệ số a, b và c như sau:

```
float a,b,c;
float delta,x1,x2;
if (a==0)
{
    if (b==0)
    {
        if (c==0)
            printf("Phương trình suy biến thành  $0x^2+0x+0=0$ , và có vô số nghiệm!");
        else
            printf("Phương trình suy biến thành  $0x^2+0x+c=0$ , vô nghiệm!");
    }
    else
    {
        printf("Phương trình suy biến thành  $0x^2+bx+c=0$ , có nghiệm duy nhất  $x=-c/b$ ");
    }
}
else
{
    delta = b*b - 4*a*c;
    if (delta < 0)
        printf("Phương trình vô nghiệm!");
    else if (delta == 0)
        printf("Phương trình có nghiệm kép  $x_1=x_2=-b/2a$ ");
    else
        printf("Phương trình có hai nghiệm phân biệt  $x_1=(-b+sqrt(delta))/2a$  và  $x_2=(-b-sqrt(delta))/2a$ ");
}
```

```

delta=b*b-4*a*c;
n u delta<0
    printf("Phương trình bậc 2 có <0, vì vậy phương trình vô nghiệm!");
còn n u delta==0
    printf("Phương trình bậc 2 có =0, có nghiệm kép x1=x2=%8.2f",-b/(2*a));
n g c l i //t c >0, phương trình bậc 2 có 2 nghiệm phân biệt
{
    x1=(-b-sqrt(delta))/(2*a);
    x2=(-b+sqrt(delta))/(2*a);
    printf("Phương trình có 2 nghiệm phân biệt: x1=%8.2f và x2=%8.2f",x1,x2);
}
}

```

5) Viết chương trình nhập vào điểm Toán, Lý và Hóa của một sinh viên. Sau đó tính và xuất ra màn hình điểm trung bình cùng kết quả xếp loại của sinh viên này dựa trên:

dtb<4: Loại F

4<=dtb<5.5: Loại D

5.5<=dtb<7: Loại C

7<=dtb<8.5: Loại B

8.5<=dtb<=10: Loại A

**Hàng đầu:** sử dụng câu lệnh if (dùng ý hoặc chuyển)

```

float t,l,h,dtb;
//Nhập vào điểm 3 môn toán, lý và hóa
dtb=(t+l+h)/3;
if(dtb<4)
    printf("Điểm trung bình là %8.2f, xếp loại F",dtb);
else if(4<=dtb && dtb<5.5)
    printf("Điểm trung bình là %8.2f, xếp loại D",dtb);
else if(5.5<=dtb && dtb<7)
    printf("Điểm trung bình là %8.2f, xếp loại C",dtb);
else if(7<=dtb && dtb<8.5)
    printf("Điểm trung bình là %8.2f, xếp loại B",dtb);
else
    printf("Điểm trung bình là %8.2f, xếp loại A",dtb);

```

6) Viết chương trình nhập vào một tháng trong năm và xuất ra số ngày của tháng đó.

**Hàng đầu:** tháng 1,3,5,7,8,10,12 có 31 ngày; tháng 4,6,9,11 có 30 ngày; riêng tháng 2 có 29 ngày nếu năm nhuận – còn lại là 28 ngày.

Vì số ngày phụ thuộc vào tháng và năm là các số nguyên nên chúng ta có thể dùng câu lệnh switch như sau:

```
int thang,nam;
//nhập vào tháng và năm (tháng 2 có 28/29 ngày phụ thuộc vào năm nhuận hay không)
switch(thang)
{
    case 1:
    case 3:
    case 5:
    case 7:
    case 8:
    case 10:
    case 12:
        printf("Thang %d/%d có 31 ngày",thang,nam);
        break;
    case 4:
    case 6:
    case 9:
    case 11:
        printf("Thang %d/%d có 30 ngày",thang,nam);
        break;
    default:
        if(nam%4==0 && nam%100!=0)
            printf("Thang %d/%d có 29 ngày",thang,nam);
        else
            printf("Thang %d/%d có 28 ngày",thang,nam);
}
```

**7) Viết chương trình nhập vào một số thực dương x (kiểm tra nếu  $x < 0$  thì nhập lại và lặp lại vì tính này cho đến khi  $x > 0$ ). Sau đó xuất ra giá trị x và nhập**

**Hướng dẫn:** sử dụng câu lệnh goto hoặc câu lệnh while/do...while

Cách 1: câu lệnh goto

```
float x;
nhập:
    printf("Nhập vào giá trị của n : ");
    scanf("%f",&x);
    if(x<=0)
    {
```

```
printf("Ban da nhap sai, nhap lai!!! ");
goto nhap;
}
```

Cách 2: câu lệnh while

```
float x;
printf("Nhap vao gia tri cua n : ");
scanf("%f",&x);
while(x<=0)
{
    printf("Ban da nhap sai, hay nhap lai gia tri cua n : ");
    scanf("%f",&x);
}
```

Cách 3: câu lệnh do...while

```
float x;
do
{
    printf("Nhap vao gia tri cua n : ");
    scanf("%f",&x);
}
while(x<=0);
```

### 8) Viết chương trình nhập vào một số nguyên dương n. Tính và xuất ra màn hình giai thừa của n.

**Hướng dẫn:** trình bày chương trình mà bạn nhập vào là số dương (số dương bài 6 trên)

Sau đó sử dụng câu lệnh for/while hoặc do...while

Cách 1: dùng câu lệnh for

```
int n,i,gt;
gt=1;
for(i=2;i<=n;i++)
    gt=gt*i;
```

Cách 2: dùng câu lệnh while

```
int n,i,gt;
gt=1;
i=2;
while(i<=n)
{
    gt=gt*i;
```

```
i++;
}
```

Cách 3: dùng câu lệnh do...while

```
int n,i,gt;
gt=1;i=1;
do
{
    gt=gt*i;
    i++;
}
while(i<=n);
```

9) Viết chương trình nhập vào số nguyên dương n và tính tổng các dãy số sau:

a.  $S = 1 - 2 + 3 - 4 + \dots \pm n$

b.  $S = 1 + 1/2 + \dots + 1/n$

c.  $S = 1 + 1/2! + 1/3! + \dots + 1/n!$

**Hướng dẫn:** trình bày chương trình n>0, sau đó áp dụng công thức tính

a.  $S = 1 - 2 + 3 - 4 + \dots \pm n$

```
#include<math.h>
//khai th  i n math.h vì có s  d  i ng các hàm toán h  c nh  pow(x,y);
double S; //Khai ki  u double vì hàm pow(x,y) tr  v  ki  u double
S=0;
for(int i=1;i<=n;i++)
    S=S+pow(-1,i+1)*i;
//ho  c dùng câu l  nh if cho tr  i ng h  p i ch  n và i l
```

b.  $S = 1 + 1/2 + \dots + 1/n$

```
S=0;
for(int i=1;i<=n;i++)
    S=S+1.0/i;
//1.0/i vì chúng ta  i ang c  n l  y c  p h  n phân và p h  n nguyên trong phép chia này
```

c.  $S = 1 + 1/2! + 1/3! + \dots + 1/n!$

```
S=0;gt=1;
for(int i=1;i<=n;i++)
{
    gt=gt*i;
    S=S+1.0/gt;
}
```

10) Viết chương trình tính bình phương nhân và in ra màn hình như sau:

	1	2	3	4	5	6	7	8	9	10
1	1	2	3	4	5	6	7	8	9	10
2	2	4	6	8	10	12	14	16	18	20
3	3	6	9	12	15	18	21	24	27	30
4	4	8	12	16	20	24	28	32	36	40
5	5	10	15	20	25	30	35	40	45	50
6	6	12	18	24	30	36	42	48	54	60
7	7	14	21	28	35	42	49	56	63	70
8	8	16	24	32	40	48	56	64	72	80
9	9	18	27	36	45	54	63	72	81	90
10	10	20	30	40	50	60	70	80	90	100

**Hướng dẫn:** sử dụng các ký tự in trong C và vòng lặp for lồng nhau như sau:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    clrscr();
    printf(" %c",179);
    for(int i=1;i<=10;i++)
        printf(" %2d",i);
    printf("\n_____%c_____",179);
    //ấn phím Shift với phím dấu gạch ngang
    for(i=1;i<=10;i++)
    {
        printf("\n %c",179);
        printf("\n%2d %c",i,179);
        for(int j=1;j<=10;j++)
            printf(" %2d",i*j);
    }
    getch();
}
```

### 11) Viết chương trình giải bài toán c:

Và gà và chó

Bó l i cho tròn

36 con, 100 chân ch n

**Hướng dẫn:** trình bày giải pháp sử dụng gà và chó, sau đó dùng câu lệnh for duy trì các biến có thể có như sau:



Gà là s, chó là s →  $x > 0$  và  $y > 0$

$$\begin{cases} x+y=36 & \rightarrow x < 36 \\ 2x+4y=100 & \rightarrow 4y < 100 \rightarrow y < 25 \end{cases}$$

```
int x,y;
for(x=1;x<36;x++)
    for(y=1;y<25;y++)
        if(2*x+4*y==100 && x+y==36)
            printf("\nS gà: %d, và s chó: %d",x,y);
```

## 12) Viết chương trình giải bài toán c:

Trên trâu, trên ngựa

Trâu ngựa n n m

Trâu n m n ba

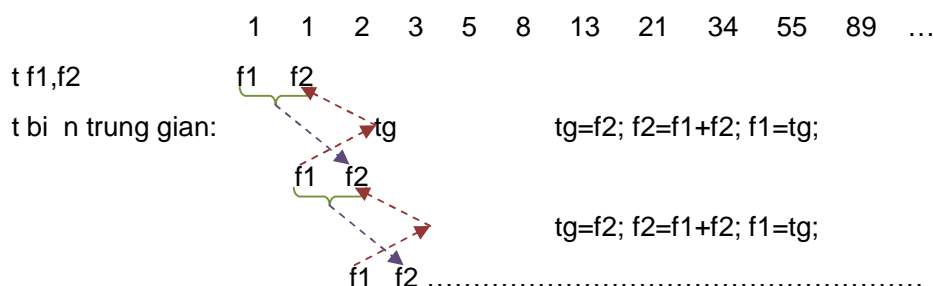
Trâu già ba con m t bố

**Hướng dẫn:** tìm nghiệm bài toán trên, tuy nhiên bị ràng buộc khi kiểm tra điều kiện có đúng:

```
if(td*5+tn*3+tg*1.0/3==100 && td+tn+tg=100)
    printf("...");
```

## 13) Viết chương trình in dãy số trong dãy Fibonacci.

**Hướng dẫn:** trên thực tế, dãy Fibonacci là dãy có định nghĩa sau:



```
int f1,f2,tg;
f1=0;
f2=1;
for(int i=1;i<=n;i++)
{
    printf("%d\t",f2);
    tg=f2;
    f2=f1+f2;
    f1=fg;
}
```

## 14) Viết chương trình nhập vào 2 số nguyên x và y. Hãy tính và xuất ra màn hình các số chung lớn nhất và bội số chung nhỏ nhất của 2 số này.

**Hướng dẫn:** phôi viết thuật toán có thể tính UCLN và BCNN cho các số âm lẫn số dương.

Thuật toán các số dương này là tính hiệu của 2 số này cho đến khi hiệu bằng 0.

Bí quyết chung nhất để tính bội tích của  $x*y$  chia cho các số chung lớn nhất ( $bcnn=x*y/ucln$ )

Chính vì vậy, cần phải tính UCLN và BCNN cho  $|x|$  và  $|y|$  (truy cập tuyệt đối).

Ví dụ  $ucln(8,4)=ucln(-8,4)=ucln(8,-4)=4$ ;

Còn nếu giá trị  $x=0$  và  $y=0$  thì kết quả ngay  $ucln=|y|$ , mà chương trình không cần tính toán gì cả

```
int x, y, a, b;
printf("Nhập vào 2 số x và y: "); scanf("%d%d",&x,&y);
a=abs(x);b=abs(y) //cần ghi lại giá trị của 2 số x,y và thao tác trên giá trị tuyệt đối
while(a!=b && a*b!=0)
    if(a>b)
        a=a-b;
    else
        b=b-a;
if(a*b==0)
    printf("Ước chung lớn nhất và bội chung nhỏ nhất của %d và %d là: %d và %d",x,y,a+b,abs(x*y)/(a+b));
else
    printf("Ước chung lớn nhất và bội chung nhỏ nhất của %d và %d là: %d và %d",x,y,b,abs(x*y)/b);
```

**15) Viết chương trình nhập vào một số nguyên. Hãy tìm các chữ số và tính tổng các chữ số của số nguyên và nhập. Sau đó, hãy xuất ra màn hình các giá trị và tính.**

**Hướng dẫn:** lấy các chữ số của một số nguyên, ta lấy từ phải sang trái (tuyệt đối hàng nghìn trở lại) như ví dụ sau:

Giả sử chỉ lấy các chữ số của  $x=2358$ .

Lần 1:  $x=2358 \rightarrow chuso=x\%10$  (tức  $chuso=2358\%10=8$ )  $\rightarrow$  lấy chữ số 8; và xem lại giá trị của các chữ số còn lại:  $x=x/10$  (tức  $x=2358/10=235$ ).

Lần 2: tiếp tục lấy lần 1 và tiếp tục lấy là 235  $\rightarrow chuso=x\%10=5$ ;  $x=x/10=23$ .

Lần 3: tiếp tục lấy  $x=23 \rightarrow chuso=x\%10=3$ ;  $x=x/10=2$ .

Lần 4: tiếp tục lấy  $x=2 \rightarrow chuso=x\%10=2$ ;  $x=x/10=0$ .

Đến khi  $x$  bằng 0 thì dừng lại và tính tổng các chữ số này.

```
int x,chuso,tong,dem;
tong=0;
dem=0;
while(x>0)
{
    chuso=x%10;
```

```
tong=tong+chuso;
dem++;
x=x/10;
}
```

**16)Viết chương trình nhập vào các tử và mẫu của hai phân số. Hãy cộng 2 phân số này và xuất kết quả ra màn hình.**

**Hướng dẫn:** giả sử 2 phân số là a/b và c/d → cần tính a/b+c/d?

```
int a,b,c,d,tu,mau;
printf("Nhập vào tử và mẫu của hai phân số : ");
scanf("%d%d%d%d",&a,&b,&c,&d);
tu=a*d+b*c;
mau=b*d;
printf("Tổng 2 phân số %d/%d + %d/%d = %8.2f",a,b,c,d,tu*1.0/mau);
```

**17)Viết chương trình nhập vào một số nguyên x. Hãy kiểm tra xem số này có thuộc dãy Fibonacci hay không và xuất câu khẳng định ra màn hình.**

**Hướng dẫn:** một số không thuộc dãy Fibonacci khi và chỉ khi số đó không là hai số liên tiếp của dãy Fibonacci. Chính vì vậy, nếu x vẫn còn lớn hơn f2 thì lặp lại việc tính f2 tiếp theo như sau:

```
int f1,f2,x;
//x nhập từ bàn phím
f1=1; f2=1;
while(x>f2)
{
    int tg=f2;
    f2=f1+f2;
    f1=tg;
}
if(x==f2)
    printf("%d thuộc dãy Fibonacci!",x);
else
    printf("%d không thuộc dãy Fibonacci!!",x);
```

**18)Viết chương trình nhập vào giá trị của một số thực x. Hãy tính và xuất ra màn hình giá trị của  $\sin(x)$  bởi:**

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots$$

với sai số cho phép.  $|x^{2n+1}/(2n+1)!| < 1E-4$

**Hướng dẫn:** từ công thức tổng quát ta có số hạng tiếp theo là:  $(-1)^i x^{2i+1}/(2i+1)!$

```
double x,tong,E;
int n,i,gt;
```

```
//nhập vào giá trị của x và E
//tính tổng sin(x)
gt=1;
tong=0;
i=0;
while(fabs(pow(x,2*i+1)/gt)<(E-4) )
{
    tong=tong+pow(-1,i)*pow(x,2*i+1)/gt;
    i++;
    gt=gt*(2*i)*(2*i+1);
}
```

## BÀI THỰC HÀNH SỐ 3

(Chương III: HÀM VÀ CẤU TRÚC CHƯƠNG TRÌNH

Chương IV: COUNTER)

Hàm là một thành phần cơ bản trong hầu hết các ngôn ngữ lập trình và C cũng vậy. Hàm được sử dụng trong một chương trình C, cần phải định nghĩa hàm đó trước khi sử dụng chúng. Có 2 cách định nghĩa và sử dụng hàm trong C:

**Cách 1:** Định nghĩa hàm trước khi gọi hàm trong chương trình của hàm main như sau

```
#include<stdio.h>
#include<conio.h>
// định nghĩa hàm max_xy(int x,int y) tính giá trị lớn nhất của 2 tham biến x và y.
int max_xy(int x,int y)
{
    return (x>y?x:y);
}
void main()
{
    int x,y;
    clrscr();
    printf("Nhập vào 2 số x và y: ");
    scanf("%d%d",&x,&y);
    //gọi hàm max_xy(x,y) tính giá trị lớn nhất của 2 số x và y và in kết quả vào
    printf("Giá trị lớn nhất của x và y là: %d",max_xy(x,y));
}
```

**Cách 2:** Định nghĩa hàm sau khi gọi hàm trong chương trình của hàm main, vì vậy cần phải có câu lệnh khai báo trước cho hàm như sau:

```
#include<stdio.h>
#include<conio.h>
//câu lệnh khai báo trước cho hàm max_xy(int x,int y)
int max_xy(int x,int y);
void main()
{
    int x,y;
    clrscr();
    printf("Nhập vào 2 số x và y: ");
    scanf("%d%d",&x,&y);
    //gọi hàm max_xy(x,y) tính giá trị lớn nhất của 2 số x và y và in kết quả vào
    printf("Giá trị lớn nhất của x và y là: %d",max_xy(x,y));
}
```

```

}
// nhúng hàm tính giá trị lớn nhất của 2 tham biến x và y.
int max_xy(int x,int y)
{
    return (x>y?x:y);
}

```

- 1) Viết chương trình nhập vào 2 số nguyên x và y. Hãy xây dựng hàm hoán đổi giá trị của 2 số nguyên này và xuất ra giá trị của chúng sau khi hoán đổi.

**Hướng dẫn:** sau khi hoán đổi, giá trị của x chính là giá trị của y trước khi hoán đổi và giá trị của y chính là giá trị của x trước khi hoán đổi.

Chương trình sinh viên thực hiện hay viết:

```

#include<stdio.h>
#include<conio.h>
void hoanvi(int x,int y)
{
    int tg;
    tg=x;
    x=y;
    y=tg;
}
void main()
{
    int x,y;
    clrscr();
    printf("Nhap vao 2 so x va y : ");
    scanf("%d%d",&x,&y);
    printf("Gia tri vua nhap: x=%d va y=%d",x,y);
    hoanvi(x,y);
    printf("\nGia tri vua nhap: x=%d va y=%d",x,y);
    getch();
}

```

Khi chạy chương trình này, gõ nhập vào x=3,y=5 thì có 2 dòng kết quả in ra màn hình trên cho x=3 và y=5. Như vậy, hàm hoán đổi đã không thực hiện được chức năng hoán đổi của nó. Lý do dẫn đến điều này là các tham số là con trỏ mà giá trị thay đổi sau khi ra khỏi hàm, còn các tham số bình thường cho dù có thay đổi trong thân hàm thì khi ra khỏi hàm cũng vẫn nhận giá trị trước khi vào hàm.

Chương trình đúng cho bài này là:

```

#include<stdio.h>

```

```
#include<conio.h>
//sử dụng con trỏ cho các tham số trong hàm để lưu giá trị sau khi hoán đổi.
void hoanvi(int *x,int *y)
{
    int tg;
    tg=*x;
    *x=*y;
    *y=tg;
}
void main()
{
    int x,y;
    clrscr();
    printf("Nhap vao 2 so x va y : ");
    scanf("%d%d",&x,&y);
    printf("Gia tri vua nhap: x=%d va y=%d",x,y);
    hoanvi(&x,&y);
    printf("\nGia tri vua nhap: x=%d va y=%d",x,y);
    getch();
}
```

- 2) Viết chương trình nhập vào một số nguyên dương n. Hãy viết hàm tính và xuất ra màn hình giai thừa của n.

**Hướng dẫn:** trỏ chỉ tới phần bộ giá trị của n khi nhập vào là đúng.

```
long int gthua(int n)
{
    if(n==0)
        return 1;
    else
        return n*gthua(n-1);
}
```

- 3) Viết chương trình nhập vào một số nguyên dương n. Hãy viết hàm tính và xuất ra màn hình số Fibonacci thứ n của dãy số Fibonacci.

**Hướng dẫn:** trỏ chỉ tới vị trí của n trong bộ giá trị của n là đúng

```
int fib_n(int n)
{
    if(n==0 || n==1)
        return 1;
```

```
else
    return fib_n(n-2)+fib_n(n-1);
}
```

- 4) Viết chương trình nhập vào 2 số nguyên a và b. Hãy viết hàm tính và xuất ra màn hình các số chung lớn nhất của 2 số trên.

**Hướng dẫn:** viết hàm tính các số chung lớn nhất của a và b.

```
int a,b;
int ucln(int a,int b)
{
    a=abs(a);b=abs(b);
    if(a==b)
        return a;
    else if(a*b==0)
        return a+b;
    else if(a>b)
        return ucln(a-b,b);
    else
        return ucln(a,b-a);
}
```

- 5) Viết chương trình xây dựng hàm tính diện tích hình tròn với bán kính R và hàm main.

**Hướng dẫn:** xây dựng hàm `dtth(int R)` và gọi hàm này trong hàm `main()` tính và in ra diện tích hình tròn với tham số R là bán kính nhập từ bàn phím.

```
float pi=3.14;
float dtth(int r) //r là tham biến trong nh ngha hàm dtth
{
    return r*r*pi;
}
void main()
{
    float R;
    printf("Nhap vao ban kinh R : ");scanf("%f",&R);
    printf("Dien tich hinh tron = %8.2f",dtth(R)); //R là giá trị truyền vào khi gọi hàm dtth
    getch();
}
```



6) Viết chương trình xây dựng hàm tính tổng các chữ số và hàm kiểm tra số nguyên tố. Sau đó hãy gọi 2 hàm này trong hàm main cho kết quả thể hiện kết quả trên màn hình.

**Hướng dẫn:** vì vị trí của chữ số và tính tổng các chữ số có thể thể hiện được cùng nhau, nên có 2 giá trị cùng một biến để lưu trữ tổng các chữ số và số chữ số. Nhưng câu lệnh return trong một hàm chỉ cho phép trả về một giá trị. Chính vì vậy, cùng một biến để lưu trữ các tham biến còn lại như 2 giá trị trả về trong hàm.

```
int tong_dem_chuso(int n,int *dem,int *tong)
{
    int chuso;
    while(n>0)
    {
        chuso=n%10;
        *tong=*tong+chuso;
        *dem=*dem+1;
        n=n/10;
    }
}

void main()
{
    int n,tong,dem;
    printf("Nhap vao so tu nhien n : ");scanf("%d",&n);
    tong_dem_chuso(n,&dem,&tong);
    printf("So %d co %d chu so va tong cac chu so = %d",n,dem,tong);
    getch();
}
```

7) Viết hàm kiểm tra một số có phải là số nguyên tố hay không?

**Hướng dẫn:** một số là số nguyên tố khi và chỉ khi nó chỉ chia hết cho 1 và chính nó.

```
int kiemtra_nto(int n)
{
    for(int i=2;i<=floor(sqrt(n));i++)
        if(n%i==0)
            break; //n không chia hết cho i, thì dừng vòng lặp for vì có thể khẳng định n là hợp số
    if(i>floor(sqrt(n)))
        return 1; // n là số nguyên tố, hàm kiểm tra trả về giá trị 1
    else
        return 0; // n là hợp số
}
```

- 8) Viết chương trình xây dựng hàm tính diện tích hình thang với các tham số như sau:

**Hướng dẫn:** Các tham số đầu vào là độ dài đáy lớn, độ dài đáy nhỏ và chiều cao. Hàm trả về giá trị diện tích.

```
void dtthang(float daylon, float daynho, float chieucao, float *kq)
{
    *kq = (daylon + daynho) * chieucao / 2;
    return;
}
```

- 9) Viết chương trình xây dựng hàm giải phương trình bậc hai với các tham số như sau:

**Hướng dẫn:** Nhập các tham số a, b, c và hai biến để lưu nghiệm.

```
void ptbac2(float a, float b, float c, float *x1, float *x2);
```

## BÀI THỰC HÀNH 4

(Chương V: DỮ LIỆU KINH LƯỢNG)

- 1) Viết chương trình nhập vào một mảng các số thực từ bàn phím. Hãy viết chương trình tìm giá trị lớn nhất của mảng và số hạng của mảng này. Tìm giá trị lớn nhất trong mảng nói trên.

**Hướng dẫn:** xây dựng hàm tìm giá trị lớn nhất của mảng số thực. Nhập vào mảng rồi gọi hàm và in ra kết quả.

```
float max_xy(float x,float y)
{
    return x>y?x:y;
}

void main()
{
    float m[100],max;
    int n;
    printf("Nhap vao so phan tu n cua mang : ");scanf("%d",&n);
    for(int i=1;i<=n;i++)
        scanf("%f",&m[i]);
    max=m[1];
    for(i=2;i<=n;i++)
        max=max_xy(max,m[i]);
}
```

- 2) Viết chương trình trong đó có các hàm: Nhập dãy số, kiểm tra dãy số tăng dần, chèn số vào vị trí thích hợp trong dãy tăng dần. Trong hàm main() yêu cầu nhập dãy số và kiểm tra. Kiểm tra nếu dãy số tăng dần thì chèn số vào vị trí thích hợp trong dãy.

```
//hàm nhập vào một mảng số thực
void nhap(int n,float a[])
{
    for(int i=1;i<=n;i++)
    {
        printf("a[%d]= ",i);
        scanf("%f",&a[i]);
    }
}

//hàm xuất mảng
void xuat(int n,float a[])
{
}
```

```

for(int i=1;i<=n;i++)
    printf("%.1f  ",a[i]);
}
//hàm kiểm tra mảng có tăng dần hay không? Nếu tăng dần thì trả về 'C' (có), ngược lại
//trả về 'K' (không)
char ktra_tangdan(float a[])
{
    for(int i=1;i<n;i++)
        for(int j=i+1;j<=n;j++)
            if(a[j]<a[i])
                break;
    if(j<=n)
        return 'C';
    else
        return 'K';
}
//hàm chèn mảng vào vị trí thích hợp: chỗ cho biết số cần chèn, và chính số quy định
//vị trí chèn trong mảng tăng dần.
void chen(float x,float a[])
{
    int vt=1;
    while(vt<=n && a[vt]<=x)
        vt++;
    for(int j=n;j>=vt;j--)
        a[j+1]=a[j];
    a[vt]=x;
}
void main()
{
    float a[100],x;
    int n;
    clrscr();
    printf("Nhập vào số phần tử n của mảng : ");scanf("%d",&n);
    nhap(n,a);
    xuat(n,a);
    if(ktra_tangdan(a)=='C')
        printf("\nMảng tăng dần!!!");
    else

```

```
printf("\nMang không tang dan!!!");
printf("\nNhập vào một số cần chen x= ");scanf("%f",&x);
chen(x,a);
xuat(n+1,a);
}
```

- 3) Viết chương trình trong đó có các hàm: Nhập ma trận, In ma trận, Cộng hai ma trận, Nhân hai ma trận. Trong hàm main() yêu cầu nhập hai ma trận A nxn, B nxn, In ra màn hình ma trận A, ma trận B, ma trận tổng, ma trận tích.

```
void tong_matran(float A[n][n],float B[n][n],float T[n][n])
{
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
            T[i][j]=A[i][j]+B[i][j];
}

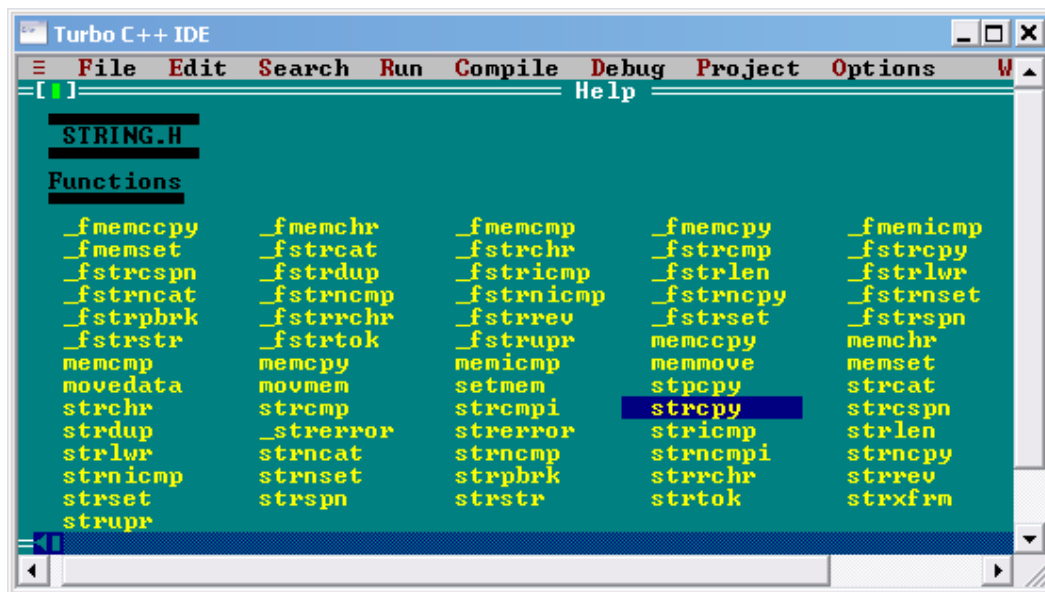
void tich_matran(float A[n][n],float B[n][n],float Ti[n][n])
{
    for(int i=1;i<=n;i++)
        for(int j=1;j<=n;j++)
        {
            float sohang=0;
            for(int k=1;k<=n;k++)
                sohang=sohang+A[i][k]*B[k][j];
            Ti[i][j]=sohang;
        }
}
```

# BÀI THỰC HÀNH SỐ 4

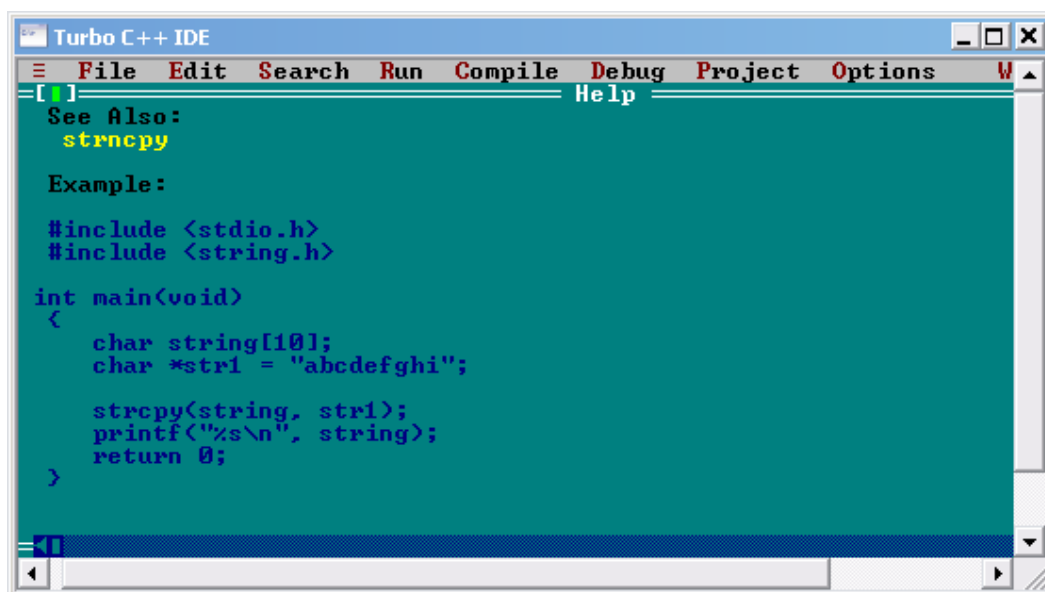
(Chương VI: DỮ LIỆU KÍ TỰ XÂU KÝ TỰ)

Trong ví dụ về dữ liệu ký tự, vì các dòng thành tạo các hàm thuộc thư viện string.h được xây dựng sẵn trong các phần mềm C là rất cần thiết.

Vào Help → chọn Index → gõ cú pháp string.h, rồi nhấn Enter.



Sau đó click chọn một hàm nào đó để xem (cấu trúc hàm và cách sử dụng hàm) và nhấn Enter. Màn hình sẽ hiển thị ra cho bạn tiếng Anh mô tả về hàm. Tuy nhiên, đa số sinh viên không hiểu tiếng Anh, nên có thể hiểu ý nghĩa và công dụng của hàm bằng cách kéo thanh công cụ xuống dưới khi nào gặp một Example như hình sau:



Sau đó, copy nội dung chương trình ví dụ trên vào cửa sổ soạn thảo C và nhấn Ctrl-F9 để chạy chương trình. Lúc này, với các giá trị nhúng vào và kết quả xuất ra màn hình sẽ hiển thị ra giúp chúng ta hiểu được chức năng – công dụng của hàm một cách nhanh chóng.

**Ví dụ về phát biểu cho các biến kiểu chuỗi ký tự trong ngôn ngữ C:**

+ Cách thứ nhất: khai báo theo mảng

**char bien[chiều dài của biến];**

→ lúc này, biến sẽ có p s bytes cho biến **bien** là **chiều dài của biến + 1** (bytes) để lưu trữ nội dung của chuỗi **bien**, trong đó byte cuối cùng để lưu trữ ký tự '\0' báo hiệu chuỗi kết thúc.

+ Cách thứ hai: khai báo theo con trỏ

**char \*bien;**

→ với khai báo này, trước tiên biến sẽ có phát 2bytes dùng để lưu trữ địa chỉ của biến **bien** (và địa chỉ của biến này). Lúc này, chúng ta có thể lưu trữ thì phải sử dụng hàm malloc() hoặc calloc() thu được từ file "alloc.h".

**Lưu ý:**

- + M i ký t c a xâu c ch a trong m t ph n t c a m ng. làm vi c v i ph n t c a xâu ký t ta g i qua tên xâu ký t và ch s nó trong xâu theo cú pháp: **Tên xâu ký t [ch s];** ch s này có thể ch y t 0 t i dài c c i c a xâu ký t . N u v trí ó n m ngoài dài th c c a xâu ký t thì ph n t ó c a xâu ký t không có giá tr xác nh → Vì v y khi làm vi c v i xâu ta ph i bi t c dài c a xâu b ng cách dùng hàm **strlen(xâu)**.
- + Vì xâu ký t là m t m ng nên C không cho phép gán hay c ng ... các xâu ký t . th c hi n vi c này ta dùng các hàm thu c th v i n string.h nh strcpy, strcat... ho c gán giá tr ngay trong lúc khai báo.

M t s hàm x lý chu i (xâu) ký t thông d ng trong th v i n string.h

- 1) **Hàm strcpy()** – sao chép toàn bộ nội dung của chuỗi nguồn vào chuỗi đích  
**char \*strcpy(char \*Des, const char \*Source)**

**Ví dụ:** char hoten[30];

```
strcpy(hoten, "Nguyen Van A");           //gán cho biến hoten chuỗi ký tự "Nguyen Van A"
//không cần gán: hoten="Nguyen Van A";
```

**Lưu ý:** không cần sử dụng phép gán chuỗi ký tự như hoten="Nguyen Van A" sau khi đã có dòng khai báo char hoten[30]; mà chỉ có thể v a khai báo v a gán giá tr nh sau:

**char hoten[30]="Nguyen Van A";**

- 2) **Hàm strcat()** – c ng 2 chuỗi ký tự  
**char \*strcat(char \*des, const char \*source)**

Hàm này có tác dụng ghép chuỗi nguồn (source) vào chuỗi đích (des), tức là sau khi gọi hàm strcat(dich,nguồn) thì hàm sẽ ghép chuỗi nguồn vào phía sau chuỗi đích rồi gán chuỗi v a ghép này cho chuỗi đích.

**Ví dụ:** char dich[30],nguồn[30];

```
strcpy(dich, "Nguyen Van A");           //t c là gán chuỗi ký tự "Nguyen Van A" vào biến dich
strcpy(nguồn, " hoc lop 08T2");         //t c là gán chuỗi ký tự " hoc lop 08T2" vào biến nguồn
strcat(dich,nguồn);                     //lúc này, chuỗi dich="Nguyen Van A hoc lop 08T2";
```

**Lưu ý:** không cần sử dụng phép c ng nh dich=dich + nguồn; (gi ng nh ki u d li u s ã h c)

**3) Hàm strlen() – xác định độ dài của chuỗi**

int strlen(const char\* s)

Hàm này dùng để xác định độ dài của chuỗi ký tự.

**Ví dụ:** char hoten[30];

strcpy(hoten, "Nguyen Van A");

strlen(hoten)=12;

//chuỗi hoten trên có chiều dài là 12 ký tự

**4) Hàmstrupr() – in chuỗi ký tự thành chuỗi ký tự hoa**

char \*strupr(char \*s)

**Ví dụ:** char hoten[30];

strcpy(hoten, "Nguyen Van A");

strupr(hoten);

//hoten="NGUYEN VAN A";

**Lưu ý:** nếu muốn in chuỗi ký tự thành chuỗi ký tự hoa thì dùng hàm toupper(char c) trong thư viện ctype.h

**5) Hàmstrlwr() – in chuỗi ký tự thành chuỗi ký tự thường**

char \*strlwr(char \*s)

**Ví dụ:** char hoten[30];

strcpy(hoten, "Nguyen Van A");

strlwr(hoten);

//hoten="nguyen van a";

**6) Hàmstrncpy() – sao chép một phần chuỗi, có thể là n ký tự hoặc đến cuối chuỗi nếu n sang chuỗi rỗng**

char \*strncpy(char \*Des, const char \*Source, size\_t n)

**Ví dụ:** char hoten[30], ho[30];

strcpy(hoten, "Nguyen Van A");

strncpy(ho, hoten, 6); // lấy 6 ký tự từ biến hoten trên (chính là chuỗi "Nguyen"), gán cho biến ho

**7) Hàm strchr() – trích một phần chuỗi (nếu có thì trả về một chuỗi, ngược lại trả về NULL)**

char \*strchr(const char \*str, int c)

Trong đó, c chính là giá trị mã ASCII của một ký tự, ví dụ c=97 có nghĩa là trích chuỗi ký tự 'a' hoặc nếu c=101 thì c là trích chuỗi ký tự 'e'.

**Ví dụ:** char hoten[30]="Nguyen Van A", st\_trich[30];

strcpy(st\_trich, strchr(hoten, 101));

//trích chuỗi bắt đầu từ ký tự 'e' (có mã ASCII 101) từ chuỗi hoten và gán vào cho biến st\_trich

//st\_trich nhận giá trị là "en Van A"

**8) Hàm strstr() – tìm kiếm nội dung chuỗi (nếu có thì trả về một chuỗi, ngược lại trả về NULL), có thể là tìm kiếm xuất hiện đầu tiên của một chuỗi st2 trong chuỗi st1 và vị trí pháp giao là**

char \*strstr(const char \*st1, const char \*st2)



**Ví dụ :** char st1[30],st2[30],ketqua[30];

strcpy(st1,"Xin chao cac ban");

strcpy(st2,"chao");

strcpy(ketqua,strstr(st1,st2)); //chu i ketqua nh n giá tr là "chao cac ban"

- 9) Hàm strcmp() – so sánh chuỗi, kết quả trả về là m t s nguyên có giá tr b ng hi u mã Ascii c a st1 và st2 t i v trí u tiên x y ra s khác nhau (st1<st2 n u k t qu là s âm, st1=st2 n u k t qu b ng 0, và st1>st2 n u k t qu là s d ng)**

int strcmp(const char \*st1,const char \*st2)

**Ví dụ :** char st1[30]="xin chao",st2[30]="xin loi";

int ketqua;

ketqua=strcmp(st1,st2); // -9

//do 2 chuỗi st1 và st2 khác nhau t v trí th 5, t c là ký t 'c' chuỗi st1 và ký t 'l' chuỗi

//st2, mà mã Ascii c a 'c'=99 và 'l'=108 nên ketqua=99-108=-9<0, do ó st1<st2

- 10) Hàm stricmp() – so sánh chuỗi mà không phân bi t ký t th ng hay hoa, và k t qu tr v t ng t k t qu tr v c a hàm strcmp()**

int stricmp(const char \*st1,const char \*st2)

**Ví dụ :** char st1[30]="Xin Chao",st2[30]="xin loi";

int ketqua;

ketqua=stricmp(st1,st2); // -9

- 11) Hàm memset() – kh i t o chuỗi, c th là kh i t o b ng cách t n ký t u tiên c a chuỗi là ký t c**

memset(char \*Des,int c,size\_t n)

**Ví dụ :** char matkhau[30];

memset(matkhau,'\*',8); //matkhau="\*\*\*\*\*"

- 12) Các hàm atoi(), atof(), atol() – i chuỗi ra s (thu c th vi n stdlib.h), n u thành công thì k t qu tr v là các s nguyên, ng c l i k t qu tr v là 0**

int atoi(const char \*st);

long atol(const char \*st);

float atof(const char \*st);

**Ví dụ :** char soint="12",solong="12",sofloat="12.8";

int soi;

long int sol;

float sof;

soi=atoi(soint); //soi=12 có giá tr s là 12 (còn soint="12" – là m t chuỗi ký t "12")

sol=atol(solong); //sol=12 có giá tr s nguyên dài là 12

sof=atof(sofloat); //sof=12.8 có giá tr s th c là 12.8

**L u ý:** chuỗi "12" khác v i s nguyên có giá tr là 12.

13) Các hàm `tolower()` và `toupper()` thuộc thư viện `ctype.h` giúp biến ký tự thành ký tự thường và ký tự hoa (khác với các hàm `strupr()` và `strlwr()` là biến đổi chuỗi ký tự)

**Ví dụ:** `char c='A',ch='b';`

`c=tolower(c);`

`//c='a';`

`ch=toupper(ch);`

`//ch='B';`

**Bài tập:**

1) Hãy viết chương trình nhập vào một chuỗi ký tự, thể hiện vị trí của chuỗi thành chuỗi hoa và viết hàm `Doi_Dau_Tu()`. Xuất ra màn hình chuỗi trước và sau khi gọi hàm.

**Hướng dẫn:** trước khi thêm ký tự trống ' ' vào phía trước chuỗi quy luật áp dụng cho chuỗi

sử dụng hàm `for` cho biến `i` chạy từ `i=0` đến `i+1<=len(st)-1` và thể hiện:

Kiểm tra `u[st[i]]=' '` và `st[i+1]!=' '` thì

Chuyển `st[i+1]` thành chuỗi hoa bằng hàm `toupper()`

2) Viết chương trình nhập vào một chuỗi và một từ. Cho biết tổng số xuất hiện trong chuỗi bao nhiêu lần?

**Hướng dẫn:** gọi chuỗi nhập vào là `str` và từ cần tìm là `tu`. tìm kiếm bằng hàm tìm kiếm chuỗi `strstr(str,tu)`. Tuy nhiên, có thể `tu` xuất hiện nhiều lần trong chuỗi nhờ hàm `strstr()` chuyển vị trí đầu tiên của chuỗi. Chính vì vậy, cần phải dùng vòng lặp `while/do...while` với điều kiện `strstr(str,tu)!=NULL` thì tiếp tục tìm, ngược lại sau khi tìm xong thì chuỗi `str` phải chuyển tiếp đến vị trí tiếp theo.

**Ví dụ:** `char str[80]="chao co chao thay chao chu chao bac chaochaochao",tu[4]="chao";`

//nhập vào, số lần xuất hiện của "chao" trong chuỗi `str` là 7 lần.

//nhận dùng hàm `strstr(str,tu)` thì chỉ tìm thấy vị trí đầu tiên. Chính vì vậy, sau khi tìm xong

//mặt thì cần phải chuyển vị trí tiếp theo để tìm kiếm tiếp trong `str`, cần là `str+=strlen(tu)` để chuyển tiếp

//chuỗi "chao bac" bằng cách chuyển ký tự kết thúc chuỗi '\0' về vị trí thích hợp

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
    char str[80],tu[4],tam[80];
    int dem=0,n;
    clrscr();
    printf("Nhap vao chuoi ky tu : ");gets(str);
    printf("\nNhap vao tu can tim : ");gets(tu);
    n=strlen(tu);
    strcpy(tam,str);
```

```
while(strstr(tam,tu)!=NULL)
{
    dem++;
    strcpy(tam,strstr(tam,tu));
    //tam[strlen(tam)]='\0';
    strrev(tam);
    tam[strlen(tam)-n]='\0';
    strrev(tam);
}
printf("\nTu %s xuất hiện trong chuỗi \"%s\" : %d lần",tu,str,dem);
getch();
}
```

3) Vì thế trình nh p vào m t chu i ký t và ki m tra xem chu i ó có i x ng hay không?

**H ng d n:** so sánh chu i nh p vào và chu i ó c a nó có b ng nhau hay không?

## BÀI THỰC HÀNH 4

(Chương VII: DỮ LIỆU NHẸNH A – DỮ LIỆU KÍ U C U TRÚC)

- Kiểm tra liên tục: dùng từ khóa enum và không cần ghi trọng tâm các giá trị của nó.

**Ví dụ:** enum mausac{xanh,tim,vang,do,cam,luc};

```
enum mausac mau; //khai báo biến mau thuộc kiểu enum mausac đã nhúng trên
if(mau==tim)
    printf....
```

- nhúng kiểu: dùng từ khóa typedef

**Ví dụ:** typedef int songuyen; //tổng quát kiểu dữ liệu mới có tên là songuyen giống kiểu dữ liệu int songuyen x;

- Kiểm tra liên tục: có thể viết nhúng và khai báo biến thuộc kiểu cấu trúc ngay phía sau định nghĩa }. Và từ khóa typedef có thể thay thế struct hoặc không.

**[typedef] struct tênkiểu**

```
{
    //khai báo các thành phần của cấu trúc;
}[biến];
```

Cập nhật [] là có thể có hoặc không có thành phần này.

**Ví dụ:** typedef struct svien

```
{
    char hoten[40],quequan[40];
    int namsinh;
};
svien sv;
```

- Viết chương trình nhập vào một chuỗi ký tự và mô tả nội dung hiển thị của nó.**

**Hướng dẫn:** nhúng một kiểu dữ liệu cấu trúc có thành phần thứ nhất là kiểu ký tự và thành phần thứ hai là kiểu int chứa số lượng ký tự trong chuỗi, rồi khai báo một mảng các phần tử thuộc cấu trúc và nhúng như sau:

```
typedef struct KyTu_SoLuong
{
    char kt;
    int sluong;
};
KyTu_SoLuong mang[100];
```

- Viết chương trình nhập vào một danh sách các sinh viên, trong đó mỗi sinh viên có họ tên, quê quán, năm sinh và điểm toán – lý – hóa. Hãy tính và xuất ra danh sách sinh viên với điểm trung bình của mỗi sinh viên để đăng ký danh sách.**

**Hạng d n:** nh ngh a ki u d li u c u trúc nh sau:

```
struct SinhVien
```

```
{
```

```
    char hoten[40],quequan[40];
```

```
    int namsinh;
```

```
    float toan,ly,hoa,dtb;
```

```
};
```

```
SinhVien sv[100];
```

## BÀI THỰC HÀNH 4

(Chương VIII: DỮ LIỆU KIỂU FILE)

- Thông tin cần khai báo biến con trỏ kiểu file để quản lý vùng nhớ của file và khai báo các biến như sau:

FILE \*fp; (stdio.h)

- Câu lệnh fflush: nếu bạn ghi mà gọi hàm fflush thì mới thông tin từ vùng nhớ của vào file và vùng nhớ trở về trạng thái sẵn sàng; ngược lại nếu bạn gọi hàm fflush thì vùng nhớ của bạn sẽ làm sẵn sàng.
- Mã LF=10 (bao gồm 2 mã CR=13 và LF=10) → ghi tiếp xuống file
- Đọc file ra bên ngoài thì cần luôn 2 ký tự CR và LF như bạn sẽ lấy ra và chuyển thành ký tự LF

⇒ Khi đọc trên file, nếu gọi ký tự có mã 26 thì có nghĩa là kết thúc file (biểu thức cho kết thúc file là EOF=-1)

- Các thao tác trên tệp tin:
  - + Khai báo biến tệp tin (con trỏ tệp tin)
  - + Mở tệp tin bằng hàm fopen()
  - + Thực hiện các thao tác xử lý dữ liệu bằng các hàm đọc/ghi dữ liệu
  - + Đóng tệp tin bằng hàm fclose()
- Mở file: cú pháp như sau

**Biến file = fopen(tên file trên đĩa, kiểu xử lý file);**

Trong đó, tên file trên đĩa có thể là một chuỗi ký tự hay một biến kiểu chuỗi; kiểu xử lý file thu được như sau

Kiểu xử lý		Ý nghĩa
File văn bản	File nhị phân	
“r”	“rb”	Mở file để đọc
“w”	“wb”	Mở file để ghi
“a”	“ab”	Mở file đã có và ghi thêm dữ liệu tiếp nối vào cuối file này. Nếu file này chưa có thì mới tạo file mới.
“r+”	“r+b”	Mở file đã có và cho phép đọc, ghi
“w+”	“w+b”	Mở file để cho đọc và ghi.
“a+”	“a+b”	Mở file đã có hoặc tạo file mới và ghi thêm dữ liệu tiếp nối vào cuối file này hoặc

Nếu mở tệp thành công thì biến con trỏ sẽ trả về giá trị khác NULL, ngược lại (không thành công) thì con trỏ nhận giá trị NULL.

- Đóng file: cú pháp như sau

fclose(biến file);

Mục đích, nhiệm vụ kết thúc chương trình mà không có file thì file vẫn có thể chạy.

- Ghi nhớ ký tự lên file:

+ File văn bản:

```
putc(char ch, FILE bincontr);
fputc(char ch[30], FILE bincontr);
fprintf(FILE bincontr, ...);
```

+ File nhị phân:

```
fwrite(&i, sizeof(int), 1, bincontr); // ghi số nguyên vào file
```

- &i: địa chỉ biến i, đây là địa chỉ biến i
- Tiếp theo là kích thước của khối dữ liệu cần ghi vào, đây do ta chỉ ghi 1 số nguyên nên kích thước của nó là sizeof(int)
- Tiếp theo là số khối dữ liệu mà biến cần ghi vào tệp, đây là 1 khối
- Cuối cùng là con trỏ cấu trúc tệp

```
fread(&i, sizeof(int), 1, bincontr);
```

- Lưu ý khi đọc file: kiểm tra xem có hết tệp chưa? Tệp là phi khác feof(bincontr).

```
Fseek(bincontr, No * kích_thuoc_1_phan_t, SEEK_SET);
```

- Đọc ký tự từ file và lưu vào biến:

```
getc(bienfile);
fgetc(str, bienfile);
fscanf();
```

Hàm feof(bincontr) cho biết đã gặp ký tự kết thúc file hay chưa? Nếu trả về 0 thì chưa, còn trả về giá trị khác 0 (0) thì có nghĩa là kết thúc file

- Hàm viết file văn bản:

```
fprintf, fscanf, fgets, fputs
fprintf(bincontr, chuoi_dieu_khi_n, danh_sach_cac_tham_s);
→ giống như printf nhưng đưa dữ liệu ra file.
fscanf(bincontr, chuoi_cac_diet, danh_sach_dia_chi_cac_bien);
→ giống như scanf nhưng đưa dữ liệu nhập file.
```

- Con trỏ tệp tin: khi mở tệp tin cần mở ra làm việc, tìm kiếm thì mới có mở tệp tin mà thôi. Vì vậy cần ghi thông tin số xảy ra (có nghĩa là có mở tệp tin đang chạy và tên nó là con trỏ tệp tin).

### Bài tập:

- Viết chương trình mở tệp tin văn bản tên VD\_File.txt chứa trong thư mục D ghi nội dung vào tệp tin này.

Hướng dẫn: đây là tệp tin văn bản nên phải mở với chế độ "w"

```
#include<stdio.h>
#include<conio.h>
void main()
{
```

```
clrscr();
FILE *fpt;
char filename[50]="D:\\CDCNghe\\GT_CDCN\\GT_Thanh\\LT_C\\VD_File.txt";
//fpt=fopen("D:\\CDCNghe\\GiaoTrinhCDCN\\GT_Thuchanh\\LT_C\\ViDuFile.txt","w");
//Lưu ý: Tên tệp không có quá dài gì nhé dòng chú thích trên
//Bởi vì nếu bạn gõ đúng câu lệnh trên thì sẽ không thể thi được!!!
fpt=fopen(filename,"w");
if(fpt==NULL)
    printf("\n Error - Không mở được tệp!!!");
else
{
    char c;
    do
        putc(c=getchar(),fpt);
    while(c!='\n');
    //Ghi cho tệp khi gõ vào phím Enter.
    fclose(fpt);
}
//getch(); Không cần hàm getch() vì gõ phím Enter là chương trình đã thoát rồi
}
```

2) Viết chương trình mở tệp tin văn bản có tên VD\_File.txt chứa trong thư mục D ghi nội dung vào tệp tin này.

**Hướng dẫn:** Đây là tệp tin văn bản nên phải mở với chế độ “w+b” hoặc “wb”

```
#include<stdio.h>
#include<conio.h>
void main()
{
    char filename[50]="D:\\CDCNghe\\GT_CDCN\\GT_Thanh\\LT_C\\File_NhiPhan.dat";
    FILE *fpt;
    int i;
    clrscr();
    fpt=fopen(filename,"w+b");
    if(fpt==NULL)
        printf("\nError - Không mở được file!!!");
    else
    {
        for(i=1;i<=10;i++)
```



```

        fwrite(&i,sizeof(int),1,fpt);
    fclose(fpt);
}
fpt=fopen(filename,"rb");
//Mu n ghi và c file trong cùng m t ch ng trình thì ph i óng file tr c khi m c
//l i file ó
//while(fread(&i,sizeof(int),1,fpt),!feof(fpt))
// printf("\n%d",i);
do
{
    fread(&i,sizeof(int),1,fpt);
    if(!feof(fpt))
        printf("\n%d",i);
}
while(!feof(fpt));
fclose(fpt);
getch();
}

```