



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

TIN HỌC ĐẠI CƯƠNG

Phần 2. Lập trình

Bài 3. Vào ra dữ liệu trong C

Các lệnh vào ra dữ liệu

- C cung cấp 2 hàm vào ra cơ bản:
 - `printf()`
 - `scanf()`
- Muốn sử dụng 2 hàm `printf()` và `scanf()` ta cần khai báo tệp tiêu đề `stdio.h`:
 - `#include <stdio.h>`
 - hoặc
 - `#include "stdio.h"`

3.1. Hàm printf()

a. Mục đích và cú pháp:

- Mục đích:
 - Hiển thị ra màn hình các loại dữ liệu cơ bản như: Số, kí tự và chuỗi kí tự
 - Định dạng dữ liệu được hiển thị
 - Một số hiệu ứng hiển thị đặc biệt như xuống dòng, sang trang,...

a. Mục đích và cú pháp (2)

- Cú pháp:

```
printf(xau_dinh_dang [, danh_sach_tham_so] );
```

- **xau_dinh_dang**: Qui định cách thức hiển thị dữ liệu ra màn hình máy tính.
- **danh_sach_tham_so**: Danh sách các biến sẽ được hiển thị giá trị lên màn hình theo cách thức được qui định trong **xau_dinh_dang**.

a. Mục đích và cú pháp (3)

- Ví dụ:

```
#include <conio.h>
#include <stdio.h>
void main()
{   int a = 5;
    float x = 1.234;
    printf("Hien thi mot so nguyen %d và mot
so thuc %f", a, x);
    getch();
}
```

- Sẽ cho ra kết quả:

Hien thi mot so nguyen 5 va mot so thuc 1.234000

a. Mục đích và cú pháp (4)

- Trong **xau_dinh_dang** chứa:
 - Các kí tự thông thường: Được hiển thị ra màn hình.
 - Các nhóm kí tự định dạng: Xác định quy cách hiển thị các tham số trong phần **danh_sach_tham_so**.
 - Các kí tự điều khiển: Dùng để tạo các hiệu ứng hiển thị đặc biệt như xuống dòng (“\n”) hay sang trang (“\f”)...

```
printf(xau_dinh_dang [, danh_sach_tham_so]);
```

a. Mục đích và cú pháp (5)

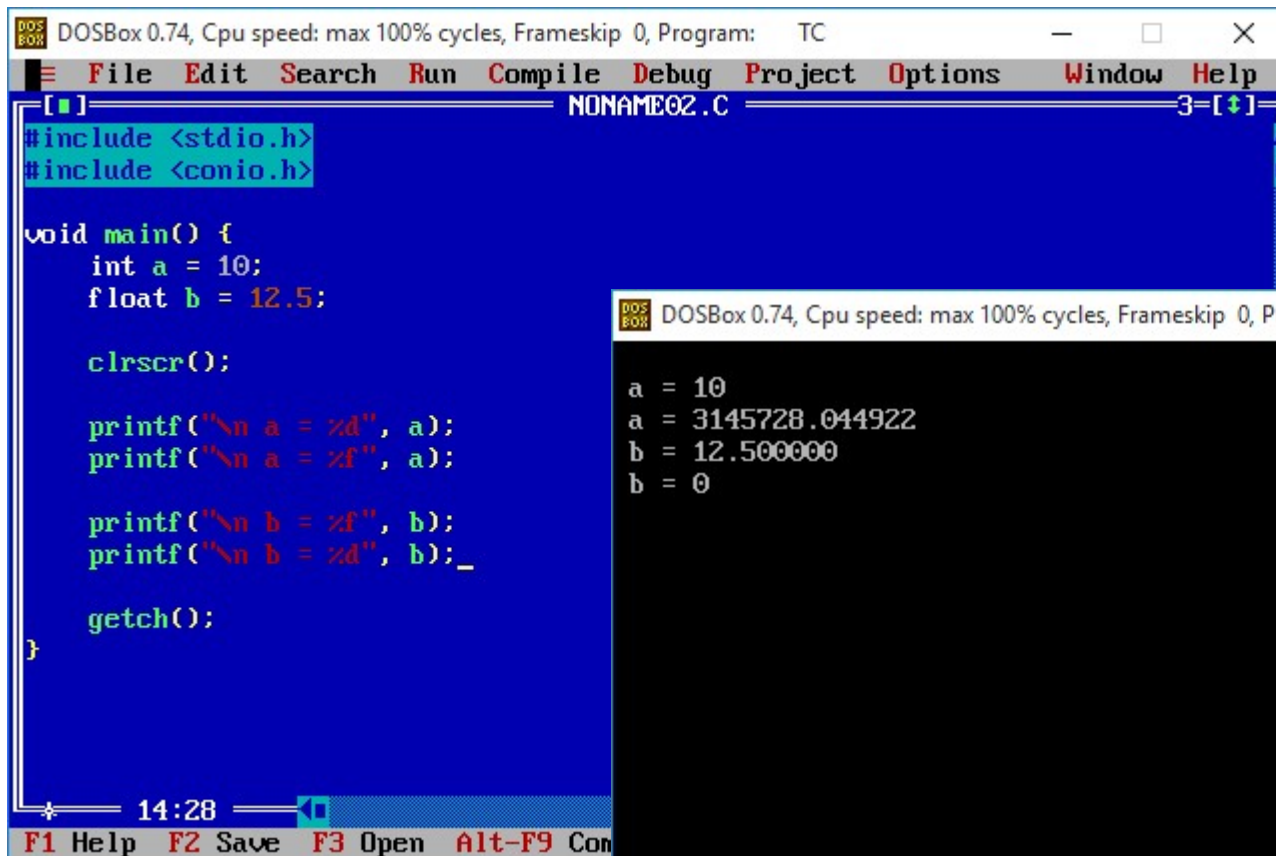
- Mỗi nhóm kí tự định dạng chỉ dùng cho một kiểu dữ liệu

Ví dụ: %d dùng cho kiểu nguyên

 %f dùng cho kiểu thực

- Nếu giữa nhóm kí tự định dạng và tham số tương ứng không phù hợp với nhau thì sẽ hiển thị ra kết quả không như ý.

Ví dụ



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

File Edit Search Run Compile Debug Project Options Window Help

NONAME02.C 3=[↑]

```
#include <stdio.h>
#include <conio.h>

void main() {
    int a = 10;
    float b = 12.5;

    clrscr();

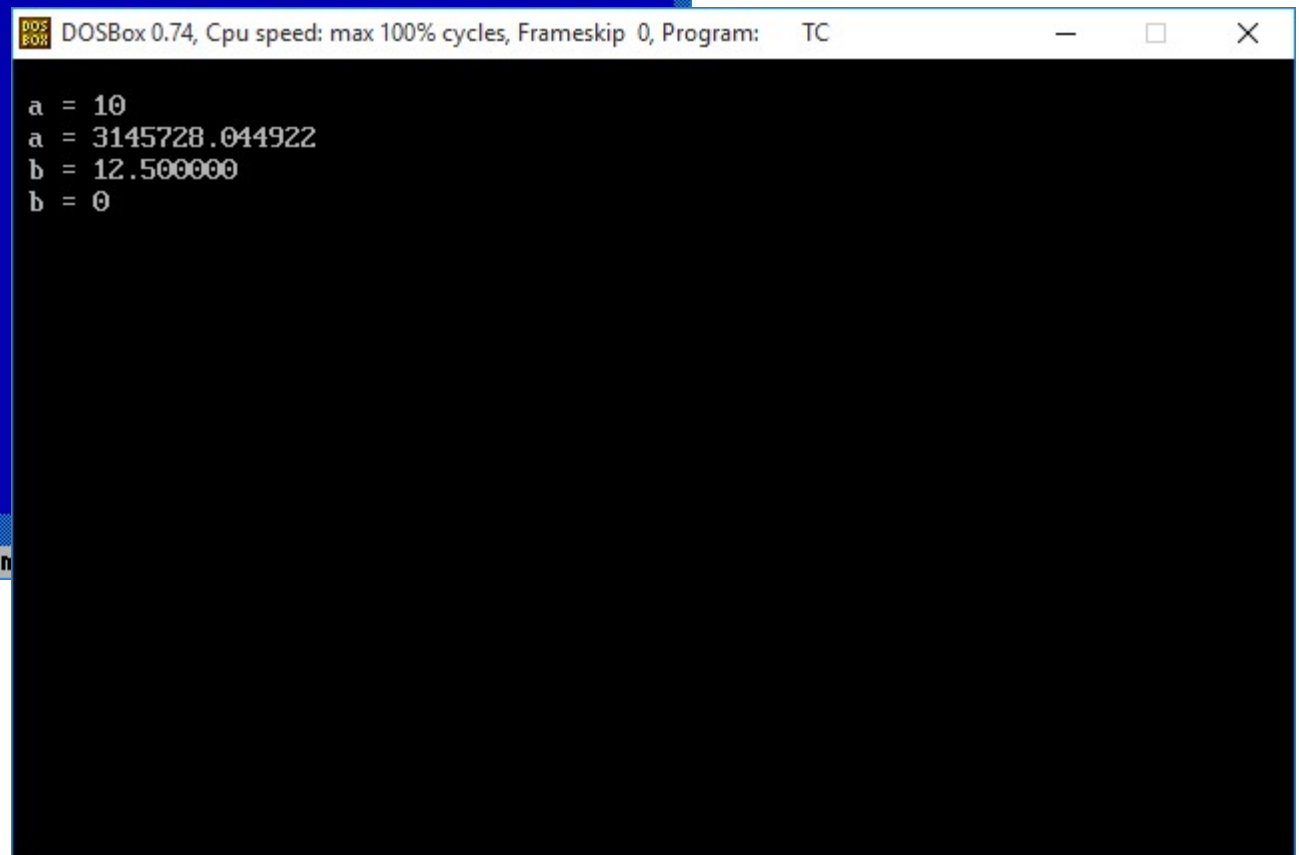
    printf("\n a = %d", a);
    printf("\n a = %f", a);

    printf("\n b = %f", b);
    printf("\n b = %d", b);_

    getch();
}
```

14:28

F1 Help F2 Save F3 Open Alt-F9 Con



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

```
a = 10
a = 3145728.044922
b = 12.500000
b = 0
```


a. Mục đích và cú pháp (6)

- **danh_sach_tham_so** phải phù hợp với các nhóm kí tự định dạng trong **xau_dinh_dang** về:
 - Số lượng
 - Kiểu dữ liệu
 - Thứ tự

b. Một số nhóm định dạng phổ biến

Nhóm kí tự định dạng	Kiểu dữ liệu	Kết quả
%c	int, char	Kí tự đơn lẻ
%i, %d	int, char	Số thập phân
%o	int, char	Số bát phân (không có 0 đằng trước)
%x, %X	int, char	Số hexa (chữ thường/chữ hoa)
%u	unsigned int/char	Số thập phân

b. Một số nhóm định dạng phổ biến (2)

Nhóm kí tự định dạng	Kiểu dữ liệu	Kết quả
%ld, %li	long	Số thập phân
%lo	long	Số bát phân (không có 0 đằng trước)
%lx, %LX	long	Số hexa (chữ thường/chữ hoa)
%lu	unsigned long	Số thập phân

b. Một số nhóm định dạng phổ biến (3)

Nhóm kí tự định dạng	Kiểu dữ liệu	Kết quả
%s	char []	Hiển thị chuỗi kí tự kết thúc bởi '\0'
%f	float/double	Số thực dấu phẩy tĩnh
%e, %E	float/double	Số thực dấu phẩy động

c. Độ rộng hiển thị - số nguyên

- Đối với **số nguyên** hoặc **ký tự** hoặc **xâu ký tự**:
 - Có dạng `%md`, với **m** là số nguyên không âm
 - Ví dụ: Có số `a = 1234`

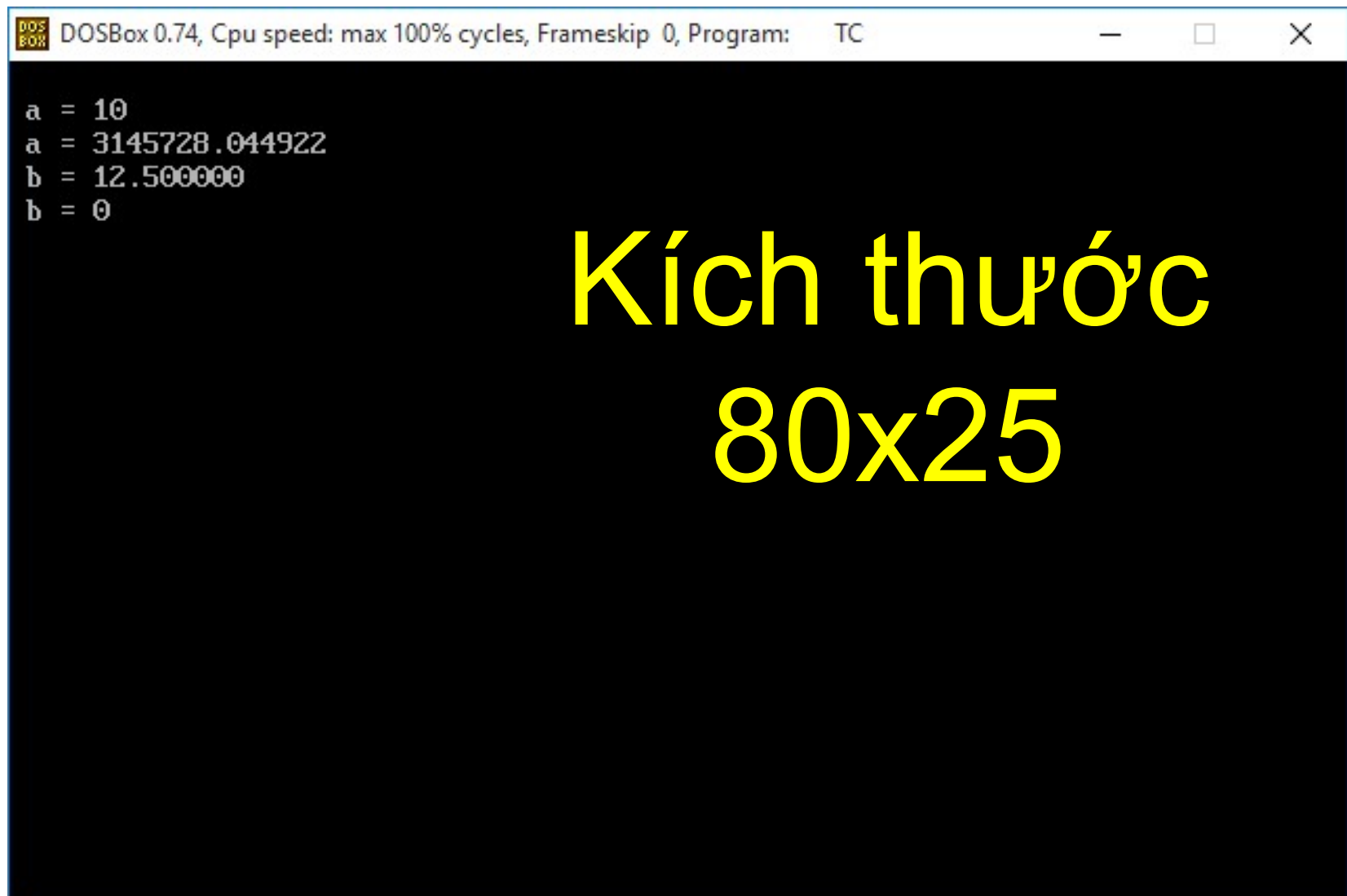
Lệnh:

```
printf("%5d", a); //đanh 5 cho de hien thi a  
printf("\n%5d", 34);
```

Cho ra kết quả: □1234
 □□□34

(□ kí hiệu cho dấu cách đơn (*space*))

Màn hình hiển thị MS-DOS



The image shows a screenshot of a DOSBox window. The title bar reads "DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC". The window contains a black terminal area with white text on the left side showing the output of a program:

```
a = 10  
a = 3145728.044922  
b = 12.500000  
b = 0
```

Overlaid on the right side of the terminal is large yellow text that reads:

Kích thước
80x25

c. Độ rộng hiển thị - số nguyên (2)

- Ví dụ:

```
printf("\n%3d %15s %3c", 1, "nguyen van a", 'g');  
printf("\n%3d %15s %3c", 2, "tran van b", 'k');
```

- Kết quả:

```
  1      nguyen van a  g  
  2      tran van b  k
```

c. Độ rộng hiển thị - số thực

- m, n là 2 số nguyên không âm

%m.nf

Trong đó:

- m vị trí để hiển thị số thực, bao gồm cả kí tự thập phân '.', và ký tự đảo dấu '-'
- n vị trí trong m vị trí đó để hiển thị phần thập phân.

c. Độ rộng hiển thị - số thực (2)

- Ví dụ:

```
printf("\n%f", 17.345);  
printf("\n%.2f", 17.345);  
printf("\n%7.2f", 17.345);
```

- Kết quả:

17.345000

17.35

□□17.35

c. Độ rộng hiển thị - Chú ý

- Khi số chỗ cần thiết để hiển thị nội dung dữ liệu lớn hơn trong định dạng:
 - Tự động cung cấp thêm chỗ mới để hiển thị chứ không cắt bớt nội dung của dữ liệu.

– Ví dụ:

```
int a = 1000;  
printf("So a la: %1d", a);
```

– Kết quả:

So a la: 1000

c. Độ rộng hiển thị - Zero padding

- Khi muốn thêm các chữ số 0 vào bên trái số hiển thị:
 - Ví dụ: mã số thí sinh thi THPT 000001, 000002, 001234, ...
 - Cách làm: thêm số 0 vào ngay sau dấu %
 - Các chữ số 0 sẽ tự động thêm vào bên trái để đủ số chỗ hiển thị
 - Ví dụ `int a = 1000;`
`printf("So a la: %06d", a);`
 - Kết quả:
`So a la: 001000`

d. Căn lề phải, lề trái

- **Căn lề phải:**
 - Khi hiển thị dữ liệu, mặc định C căn lề phải
- **Căn lề trái:**
 - Nếu muốn căn lề trái khi hiển thị dữ liệu ta chỉ cần thêm dấu trừ - vào ngay sau dấu %.

d. Căn lề phải, lề trái (2)

- Ví dụ:


```
printf("\n%-3d %-15s %.2f %-3c", 9, "nguyen  
van a", 7.5, 'g');
```

```
printf("\n%-3d %-15s %.2f %-3c", 10, "nguyen  
ha", 6.75, 'k');
```

- Kết quả:

9 nguyen van a 7.50 g

10 nguyen ha 6.75 k



The screenshot shows a Turbo C++ IDE window titled 'LE_TRAI.CPP'. The source code is as follows:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    printf("\n%-3d %-15s %.2f %-3c", 9, "nguyen van a", 7.5, 'g');
    printf("\n%-3d %-15s %.2f %-3c", 10, "nguyen ha", 6.75, 'k');
}
```

The output window, titled 'Output', shows the following results:

```
C:\TC\BIN>tc
9   nguyen van a   7.50 g
10  nguyen ha     6.75 k
```

3.2. Hàm scanf()

a. Mục đích và cú pháp:



- Mục đích:
 - Hàm **scanf()** dùng để nhập dữ liệu từ bàn phím
- Cú pháp:
`scanf(xau_dinh_dang [, danh_sach_dia_chi]);`
- Ví dụ:

```
int a; float b;  
scanf("%d%f", &a, &b);
```

a. Mục đích và cú pháp (3)



- **xâu_dinh_dang:**
 - Gồm các ký tự được quy định cho từng loại dữ liệu được nhập vào.
 - Ví dụ: với dữ liệu định nhập vào là kiểu nguyên thì xâu định dạng là : %d
- **danh_sach_dia_chi:**
 - Bao gồm các địa chỉ của các biến (toán tử &), phân tách nhau bởi dấu phẩy (,)

a. Mục đích và cú pháp (4)



- **danh_sach_dia_chi** phải phù hợp với các nhóm kí tự định dạng trong **xau_dinh_dang** về:
 - Số lượng biến cần nhập
 - Kiểu dữ liệu
 - Thứ tự

b. Một số nhóm định dạng phổ biến

Nhóm kí tự định dạng	Kiểu dữ liệu	Chú thích	Ví dụ
<code>%c</code>	char	Kí tự đơn lẻ	<code>char x; scanf("%c", &x);</code>
<code>%d</code>	int	Số thập phân	<code>int x; scanf("%d", &x);</code>
<code>%o</code>	int	Số bát phân	<code>int x; scanf("%o", &x);</code>
<code>%x</code>	int	Số hexa	<code>int x; scanf("%x", &x);</code>
<code>%u</code>	unsigned int	Số thập phân	<code>unsigned int x; scanf("%u", &x);</code>

c. Một số nhóm định dạng phổ biến (3)

Nhóm kí tự định dạng	Kiểu dữ liệu	Chú thích
%s	char[]	Nhập chuỗi kí tự kết thúc bởi '\0'
%f	float	Số thực dấu phẩy thập
%ld	long	Số nguyên
%lf	double	Số thực dấu phẩy thập

Ví dụ

```
#include <conio.h>
#include <stdio.h>
void main()
{
    // khai bao bien
    int a; float x;
    char ch; char str[30];
    // Nhap du lieu
    printf("Nhap vao mot so nguyen");
    scanf("%d", &a);
    printf("\n Nhap vao mot so thuc");
    scanf("%f", &x);
}
```

Ví dụ

```
printf("\n Nhap vao mot ki tu");  
fflush(stdin); scanf("%c", &ch);  
printf("\n Nhap vao mot xau ki tu");  
fflush(stdin); scanf("%s", str);  
  
// Hien thi du lieu vua nhap vao  
printf("\n Nhung du lieu vua nhap vao");  
printf("\n So nguyen: %d", a);  
printf("\n So thuc : %.2f", x);  
printf("\n Ki tu: %c: ", ch);  
printf("\n Xau ki tu: %s", str);  
getch();  
}
```

Ví dụ



- Kết quả:

Nhap vao mot so nguyen: 2007

Nhap vao mot so thuc: 17.1625

Nhap vao mot ki tu: b

Nhap vao mot xau ki tu: ngon ngu lap
trinh C

Nhung du lieu vua nhap vao

So nguyen: 2007

So thuc: 17.16

Ki tu: b

Xau ki tu: ngon

```

1 // SCANF.CPP
2 void main()
3 {
4     // khai bao bien
5     int a; float x;
6     char ch; char str[30];
7     // Nhap du lieu
8     printf("Nhap vao mot so nguyen: ");
9     scanf("%d",&a);
10    printf("\n Nhap vao mot so thuc: ");
11    scanf("%f",&x);
12    printf("\n Nhap vao mot ki tu: ");
13    fflush(stdin); scanf("%c",&ch);
14    printf("\n Nhap vao mot xau ki tu: ");
15    fflush(stdin); scanf("%s",str);
16
17    // Hien thi du lieu vua nhap vao
18    printf("\n Nhung du lieu vua nhap vao");
19    printf("\n So nguyen: %d",a);
20    printf("\n So thuc : %.2f",x);
21    printf("\n Ki tu: %c: ",ch);
22    printf("\n Xau ki tu: %s",str);
23 }

```

```

Output
2
Nhap vao mot so nguyen: 2007

Nhap vao mot so thuc: 17.1625

Nhap vao mot ki tu: b

Nhap vao mot xau ki tu: ngon ngu

Nhưng dữ liệu vừa nhập vào
Số nguyên: 2007
Số thực : 17.16
Ki tu: b:
Xau ki tu: ngon

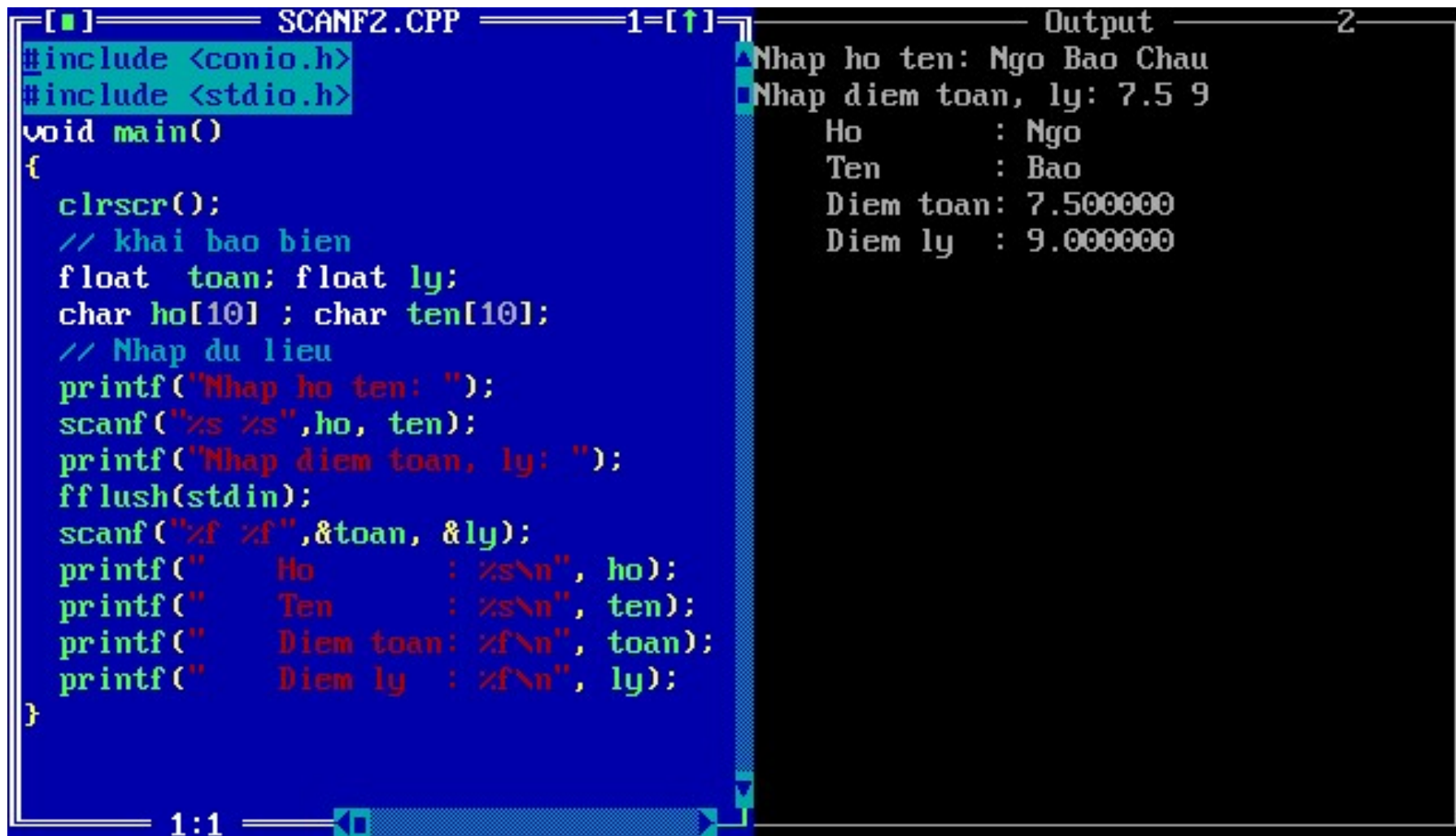
```

c. Một số quy tắc cần lưu ý



- Quy tắc 1: Khi đọc số (%i, %d, %f)
 - Hàm **scanf()** quan niệm rằng mọi kí tự số, dấu chấm (".") đều là kí tự hợp lệ.
 - Khi gặp các dấu phân cách như tab, xuống dòng hay dấu cách (space bar) thì **scanf()** sẽ hiểu là kết thúc nhập dữ liệu cho một số

c. Một số quy tắc cần lưu ý (tiếp)



The screenshot displays a C++ program in a code editor and its output in a separate window. The code, titled 'SCANF2.CPP', includes `<conio.h>` and `<stdio.h>`. It defines a `main` function that clears the screen, declares variables for total score (`toan`), average score (`ly`), and name (`ho`, `ten`). It prompts the user to enter the name and scores, then prints the results with formatted output.

```
#include <conio.h>
#include <stdio.h>
void main()
{
    clrscr();
    // khai bao bien
    float toan; float ly;
    char ho[10]; char ten[10];
    // Nhap du lieu
    printf("Nhap ho ten: ");
    scanf("%s %s", ho, ten);
    printf("Nhap diem toan, ly: ");
    fflush(stdin);
    scanf("%f %f", &toan, &ly);
    printf("    Ho      : %s\n", ho);
    printf("    Ten     : %s\n", ten);
    printf("    Diem toan: %f\n", toan);
    printf("    Diem ly  : %f\n", ly);
}
```

The output window shows the program's execution with the following text:

```
Nhap ho ten: Ngo Bao Chau
Nhap diem toan, ly: 7.5 9
Ho      : Ngo
Ten     : Bao
Diem toan: 7.500000
Diem ly  : 9.000000
```


c. Một số quy tắc cần lưu ý (tiếp)



- Quy tắc 2: Khi đọc kí tự (%c):

Hàm **scanf()** cho rằng mọi kí tự có trong bộ đệm của thiết bị vào chuẩn đều là hợp lệ, kể cả các kí tự tab, xuống dòng hay dấu cách.



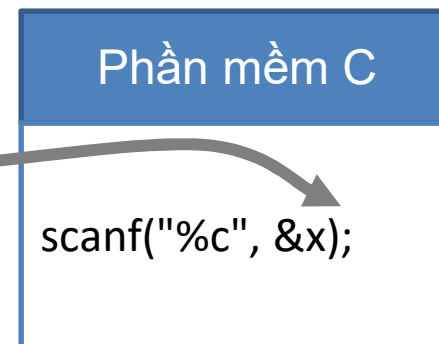
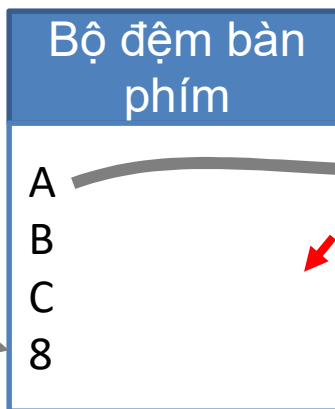
Phần mềm
(viết bằng C)

c. Một số quy tắc cần lưu ý (tiếp)



- Quy tắc 3: Khi đọc chuỗi ký tự (%s):
 - + Hàm **scanf()** nếu gặp các ký tự dấu trắng, dấu tab hay dấu xuống dòng thì nó sẽ hiểu là kết thúc nhập dữ liệu cho một chuỗi ký tự.
 - + Trước khi nhập dữ liệu ta nên dùng lệnh **fflush(stdin)** để xóa bộ đệm.

người dùng
gõ ký tự '8'



3.3. Các lệnh vào ra khác

- **Hàm `gets()` :**

Dùng để nhập vào từ bàn phím một chuỗi kí tự **bao gồm cả dấu cách**, điều mà hàm `scanf()` không làm được.

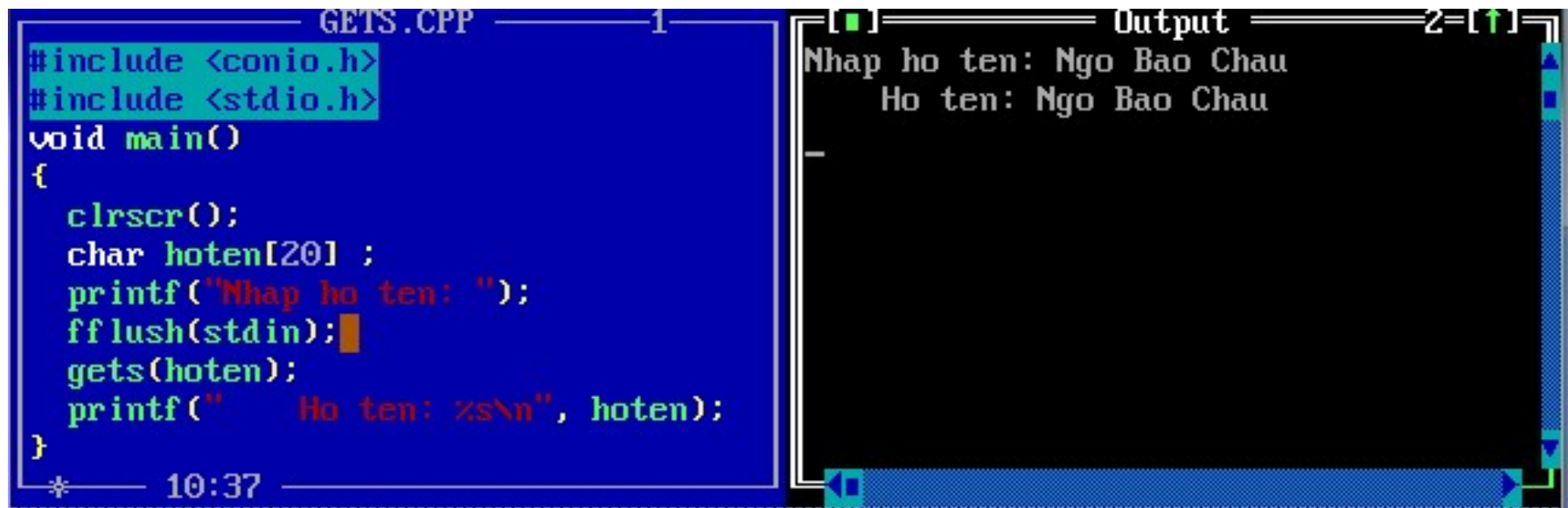
- Cú pháp :

`gets(biến_xâu_kí_tự) ;`

- Ví dụ:

```
char str[30];  
printf("Nhap vao mot xau ki tu:");  
fflush(stdin); gets(str);
```

7.6.3. Các lệnh vào ra khác



```
GETS.CPP 1
#include <conio.h>
#include <stdio.h>
void main()
{
    clrscr();
    char hoten[20] ;
    printf("Nhap ho ten: ");
    fflush(stdin);
    gets(hoten);
    printf("    Ho ten: %s\n", hoten);
}
* 10:37
```

Output

Nhap ho ten: Ngo Bao Chau
Ho ten: Ngo Bao Chau

7.6.3. Các lệnh vào ra khác (2)

- **Hàm puts():**

Hiển thị ra màn hình nội dung **xâu_kí_tự** và sau đó đưa con trỏ xuống dòng mới.

- Cú pháp:

puts (xâu_kí_tự) ;

- Ví dụ:

puts ("Nhap vao xau ki tu:");

- Tương đương với:

printf ("%s\n", "Nhap vao xau ki tu:");

Hoặc

printf ("Nhap vao xau ki tu:\n");

7.6.3. Các lệnh vào ra khác (3)

- Hàm `getch()`: thường dùng để chờ người sử dụng ấn một phím bất kì rồi sẽ kết thúc chương trình.
- Cú pháp

`getch () ;`

- Sử dụng hàm **`gets ()`** , **`puts ()`** , cần khai báo tệp tiêu đề **`stdio.h`**
- Sử dụng hàm **`getch ()`** , cần khai báo tệp tiêu đề **`conio.h`**

Ví dụ

```
#include <conio.h>
#include <stdio.h>
void main()
{
    char str[30];
    puts("Hay cho biet ho ten ban:");
    fflush(stdin); gets(str);
    printf("Xin chao ");
    puts(str);
    puts("An phim bat ki de ket thuc...");
    getch();
}
```

Ví dụ

- Kết quả:

Hay cho biet ho ten ban:

ngon ngu lap trinh C

Xin chao ngon ngu lap trinh C

An phim bat ki de ket thuc ...



The screenshot displays a code editor window titled 'PUTS.CPP' on the left and an 'Output' window on the right. The source code in the editor is as follows:

```
#include <conio.h>
#include <stdio.h>
void main()
{
    char str[30]; clrscr();
    puts("Hay cho biet ho ten ban: ");
    fflush(stdin); gets(str);
    printf("Xin chao ");
    puts(str);
    puts("An phim bat ki de ket thuc...");
    getch();
}
```

The output window shows the program's execution results:

```
Hay cho biet ho ten ban:
Ngo Bao Chau
Xin chao Ngo Bao Chau
An phim bat ki de ket thuc...
```

The status bar at the bottom of the editor shows the time as 8:20.

Bài tập

1. Viết chương trình tính diện tích và chu vi hình tròn.
2. Viết chương trình tính diện tích hình thang.
3. Viết chương trình nhập 2 số a và n. In ra màn hình giá trị bit thứ n của số a.
4. Viết chương trình nhập vào số n có 3 chữ số, in ra màn hình số n theo chiều ngược lại.
5. Viết chương trình nhập vào 1 ký tự, sau đó in ra mã ASCII của ký tự đó.

Bài tập

6. Viết chương trình thực hiện công việc sau:

- Nhập vào 3 điểm Toán, Lý, Hóa
- Tính và in ra điểm trung bình 3 môn (làm tròn đến 2 chữ số thập phân)
- Tìm và in ra điểm lớn nhất (sử dụng biểu thức điều kiện)
- Tìm và in ra điểm nhỏ nhất (sử dụng biểu thức điều kiện)