



TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

50 LỖI THƯỜNG GẶP TRONG THỰC HÀNH TIN HỌC ĐẠI CƯƠNG - IT1110

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

GIỚI THIỆU CHUNG

- Gồm 50 lỗi thường gặp khi thực hành THĐC.
- Giúp sinh viên học thực hành dễ dàng hơn.

NỘI DUNG

NỘI DUNG	LỖI	TRANG
Nộp bài trên hệ thống CodeForces	#1 - #3	5 – 7
Kiểu dữ liệu và biểu thức trong C Vào ra dữ liệu trong C	#4 - #16	9 - 21
Cấu trúc điều khiển trong C	#17 - # 35	23 - 41
Mảng và Xâu ký tự	#36 - #50	43 - 57

NỘP BÀI LÊN HỆ THỐNG CODEFORCES



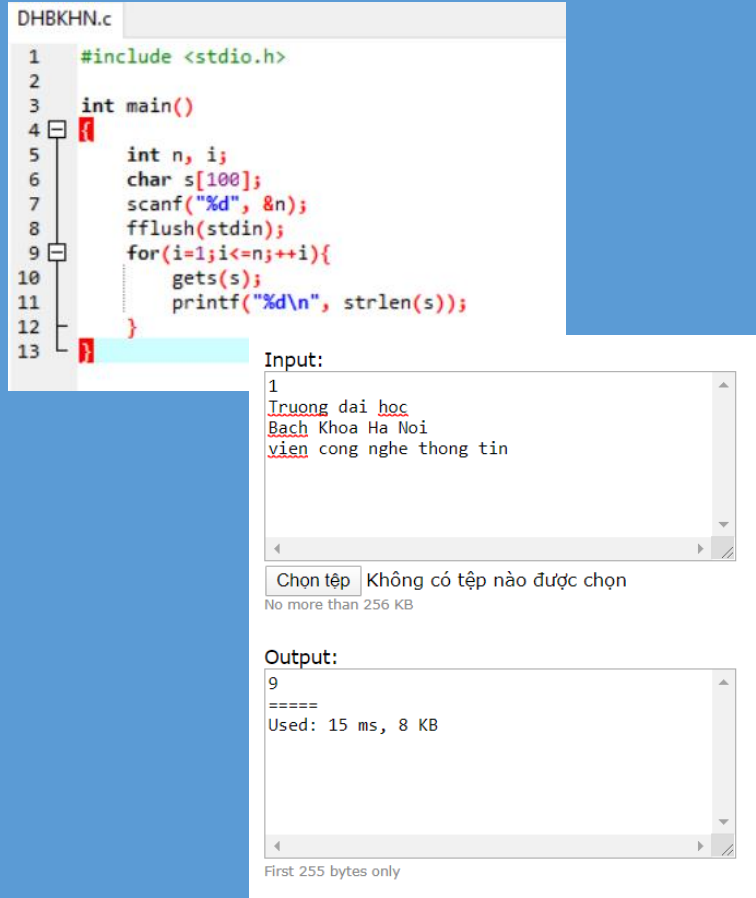
#1 – LỖI LIMIT KHI SUBMIT BÀI

LỖI XUẤT HIỆN	GIẢI THÍCH LỖI	CÁCH XỬ LÝ
<p>Lỗi limit khi submit trên hệ thống codeforces:</p> <pre>Idleness limit exceeded on test 1</pre> <pre>#include <stdio.h> #include <conio.h> int main() { //fafsf int a,b,tong; scanf("%d",&a); scanf("%d",&b); tong=a+b; printf("%d",tong); getch(); return 0; }</pre>	<p>Lỗi báo time limit ở test 1. Thường do sinh viên sử dụng các lệnh làm chương trình “đợi” hoặc thuật toán chưa tối ưu dẫn đến quá thời gian chạy. Trong ảnh bên là do dùng hàm getch().</p> <p>Hàm getch() sẽ đợi 1 input bất kỳ từ bàn phím sau đó mới chạy tiếp, vì thế, khi đọc tới lệnh này, chương trình sẽ đứng “đợi” suốt tại dòng lệnh đó.</p>	<p>Với lỗi do dùng hàm getch(): Xóa bỏ lệnh này và thư viện conio.h tương ứng.</p> <p>Với lỗi do thuật toán chưa tối ưu: kiểm tra lại và tối ưu lại thuật toán.</p>

#2 – BIÊN DỊCH LỖI DO SAI CÚ PHÁP

LỖI XUẤT HIỆN	GIẢI THÍCH LỖI	CÁCH XỬ LÝ
<pre>#include <stdio.h> int main() { int a,b,tong; scanf("%d",&a); scanf("%d",&b); tong=a+b; printf("%d",tong); }</pre> <p>Lỗi xảy ra:</p> <p>Compilation error</p>	<p>Biên dịch lỗi khi submit lên CodeForces thường do lỗi cú pháp (thiếu ngoặc, chấm phẩy...)</p>	<p>Với lỗi cú pháp, sinh viên chạy code đó trên Dev C++, trình biên dịch của Dev C++ sẽ báo vị trí lỗi.</p>

#3 – LỖI SAI OUTPUT

LỖI XUẤT HIỆN	GIẢI THÍCH LỖI	CÁCH XỬ LÝ
 <pre>DHBKHN.c 1 #include <stdio.h> 2 3 int main() 4 { 5 int n, i; 6 char s[100]; 7 scanf("%d", &n); 8 fflush(stdin); 9 for(i=1; i<=n; ++i){ 10 gets(s); 11 printf("%d\n", strlen(s)); 12 } 13 }</pre> <p>Input:</p> <p>1 Truong dai hoc Bach Khoa Ha Noi vien cong nghe thong tin</p> <p>Chọn tệp Không có tệp nào được chọn No more than 256 KB</p> <p>Output:</p> <p>9 ==== Used: 15 ms, 8 KB</p> <p>First 255 bytes only</p>	<p>Lệnh fflush(stdin) sẽ xóa bộ nhớ đệm của stdin trên codeforces. Vì thế, chương trình sẽ chạy ra output sai khi submit trên CodeForces.</p> <p>Trên máy tính, input nhập vào tuần tự, vì thế fflush chỉ xóa đi bộ nhớ tạm các ký tự đã nhập trước đó, các ký tự sau vẫn tiếp tục nhập bình thường. Còn trên Codeforces, input ở stdin đã được tạo sẵn, lệnh này sẽ xóa bỏ bộ nhớ đệm của input và làm cho chương trình chạy sai không như ý muốn.</p>	<p>Không sử dụng lệnh fflush(stdin) trong các bài phải submit lên Codeforces. Tìm hiểu các phương pháp khác để nhận được dữ liệu từ stdin cho đúng. Sinh viên có thể thử kiểm nghiệm code với chức năng custom invocation của CodeForces.</p>

Kiểu dữ liệu và Biểu thức trong C

Vào – Ra dữ liệu trong C



#4 – LỖI TRÀN SỐ

LỖI XUẤT HIỆN

checkloi.cpp

```
1  #include <stdio.h>
2  int main()
3  {
4      char c = 127;
5      c = c + 3;
6      printf("c = %d", c);
7  }
```

C:\Users\Friday\Desktop\code C\checkloi.exe

```
c = -126
-----
Process exited after 0.02209 seconds
Press any key to continue . . .
```

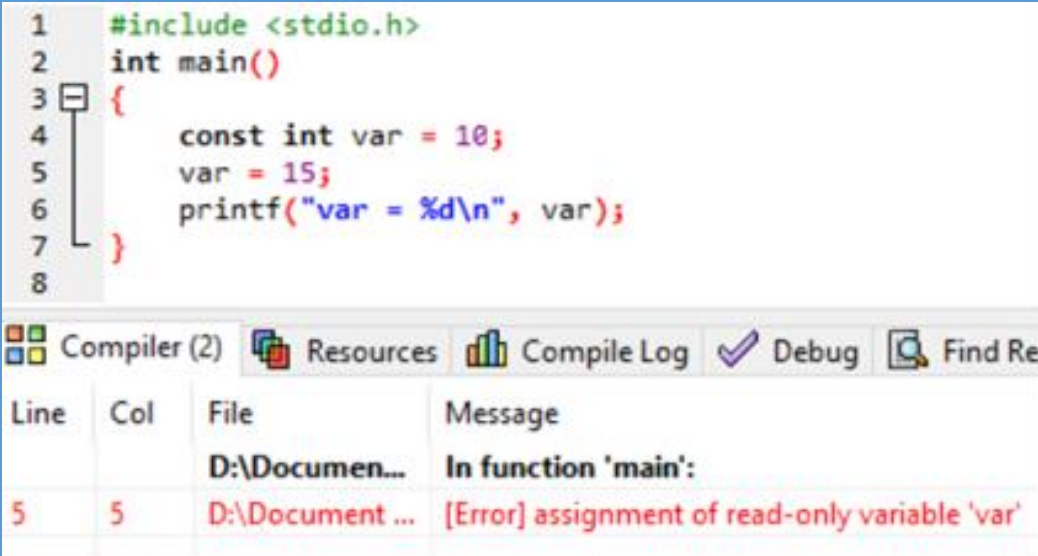
GIẢI THÍCH LỖI

Lỗi xảy ra khi thực hiện phép tính vượt quá phạm vi của kiểu dữ liệu (char) dẫn đến tràn số, tương tự cho các kiểu dữ liệu khác (chú ý với phép cộng và nhân các số lớn có thể gây tràn số)

CÁCH XỬ LÝ

Khai báo biến có kiểu dữ liệu có kích thước lớn hơn kiểu dữ liệu hiện tại chẳng hạn int, ...

#5 – LỖI KHI THAY ĐỔI GIÁ TRỊ CỦA HẲNG SỐ

LỖI XUẤT HIỆN	GIẢI THÍCH LỖI	CÁCH XỬ LÝ
 <pre>1 #include <stdio.h> 2 int main() 3 { 4 const int var = 10; 5 var = 15; 6 printf("var = %d\n", var); 7 } 8</pre> <p>The screenshot shows a C program in a compiler window. The code defines a constant integer variable <code>var</code> with the value 10. On line 5, it attempts to re-assign the value of <code>var</code> to 15. This causes a compiler error: "[Error] assignment of read-only variable 'var'". The error message is displayed in the compiler's output window at line 5, column 5.</p>	<p>Biến hằng số (constant) không được phép thay đổi.</p>	<p>Chỉ sử dụng constant với các biến cố định, không thay đổi trong suốt quá trình thực thi code.</p> <p>Với các biến có thay đổi giá trị, không được sử dụng kiểu biến constant.</p>

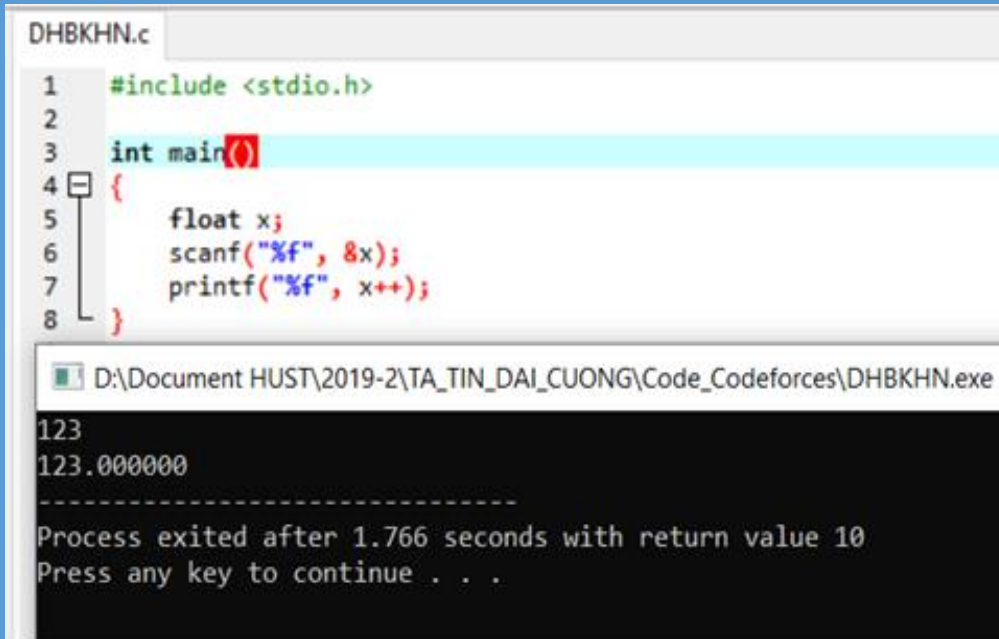
#6 – LỖI NHẦM KÝ HIỆU SO SÁNH

LỖI XUẤT HIỆN	GIẢI THÍCH LỖI	CÁCH XỬ LÝ
<pre>1 #include <stdio.h> 2 int main() 3 { 4 int a = 5, b = 4; 5 if(a = b){ 6 printf("true"); 7 } 8 else{ 9 printf("false"); 10 } 11 12 return 0; 13 }</pre> <p>C:\Users\Friday\Desktop\code C\checkloi.exe</p> <p>true</p> <p>-----</p> <p>Process exited after 0.02094 seconds with return code 0</p> <p>Press any key to continue . . .</p>	<p>Lỗi xảy ra khi thực hiện kiểm tra $a = b$ hay không. Tuy nhiên trong ví dụ này thì $b=4$ và giá trị trả về của phép gán sẽ là 4 (khác 0) nên chương trình sẽ rẽ vào nhánh $\text{if}(a=b)$ rồi in ra 'true'.</p>	<p>Thay toán tử '=' bằng toán tử '==' để kiểm tra giá trị 2 biến a và b có bằng nhau hay không.</p>

#7 – LƯU Ý KHI DÙNG “++”

LỖI XUẤT HIỆN

Nhập vào số thực x, xuất ra kết quả của x+1



```
DHBKHN.c
1  #include <stdio.h>
2
3  int main()
4  {
5      float x;
6      scanf("%f", &x);
7      printf("%f", x++);
8  }
```

D:\Document HUST\2019-2\TA_TIN_DAI_CUONG\Code_Codeforces\DHBKHN.exe

123
123.000000

Process exited after 1.766 seconds with return value 10
Press any key to continue . . .

GIẢI THÍCH LỖI

Toán tử `x++` có nghĩa là thực hiện cụm lệnh chứa `x` trước, sau đó mới tăng giá trị của `x` lên 1 đơn vị.
Ngược lại `++x` có nghĩa là tăng `x` lên trước sau đó mới thực hiện lệnh.

CÁCH XỬ LÝ

Sửa lại `x++` thành `++x` hoặc bổ sung 1 lệnh tăng `x` lên 1 đơn vị sau đó mới xuất `x` ra màn hình.

#8 – LỖI KHI ÉP KIỂU

LỖI XUẤT HIỆN

Nhập vào 2 số nguyên x và y. Xuất ra trung bình cộng của 2 số đó.

```
DHBKHN.c
1  #include <stdio.h>
2
3  int main(){
4      int x,y;
5      scanf("%d%d", &x, &y);
6      float f = (x+y)/2;
7      printf("%f", f);
8  }
```

D:\Document HUST\2019-2\TA_TIN_DAI_CUONG\Code_Codeforces\DHBKHN.exe

```
4
5
4.000000
-----
Process exited after 1.322 seconds with return value 8
Press any key to continue . . .
```

GIẢI THÍCH LỖI

Đây là lỗi sai về ép kiểu dữ liệu. Do x và y là số nguyên, nên x+y cũng là số nguyên. Mẫu số cũng là 1 số nguyên, nên kết quả của (x+y)/2 sẽ là số nguyên, sau đó giá trị nguyên này mới được gán vào biến thực float f và xuất ra màn hình.

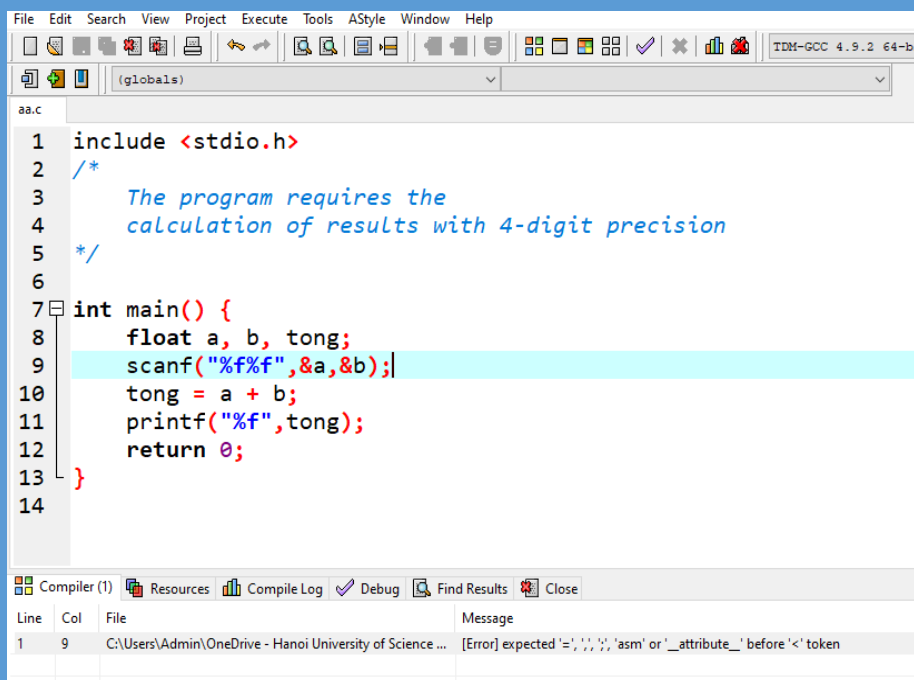
CÁCH XỬ LÝ

Thực hiện ép kiểu lại phép chia sao cho tử số hoặc mẫu số là số thực:
(float)(x+y)/2
hoặc
(x+y)/2.0

#9 – LỖI KHAI BÁO THƯ VIỆN

LỖI XUẤT HIỆN

Nhập vào 02 số thực, xuất ra tổng của 02 số thực đó với độ chính xác 4 chữ số sau dấu phẩy



```
1 include <stdio.h>
2 /*
3  The program requires the
4  calculation of results with 4-digit precision
5  */
6
7 int main() {
8     float a, b, tong;
9     scanf("%f%f",&a,&b);
10    tong = a + b;
11    printf("%f",tong);
12    return 0;
13 }
14
```

Compiler (1) Resources Compile Log Debug Find Results Close

Line	Col	File	Message
1	9	C:\Users\Admin\OneDrive - Hanoi University of Science ...	[Error] expected '=', ',', ';', 'asm' or '__attribute__' before '<' token

GIẢI THÍCH LỖI

Chương trình có lỗi cú pháp khai báo thư viện.

Ngoài ra do chương trình yêu cầu in ra đầu ra với độ chính xác 4 chữ số thì kết quả sẽ sai (%f mặc định sẽ đưa ra 6 chữ số sau dấu phẩy)

CÁCH XỬ LÝ

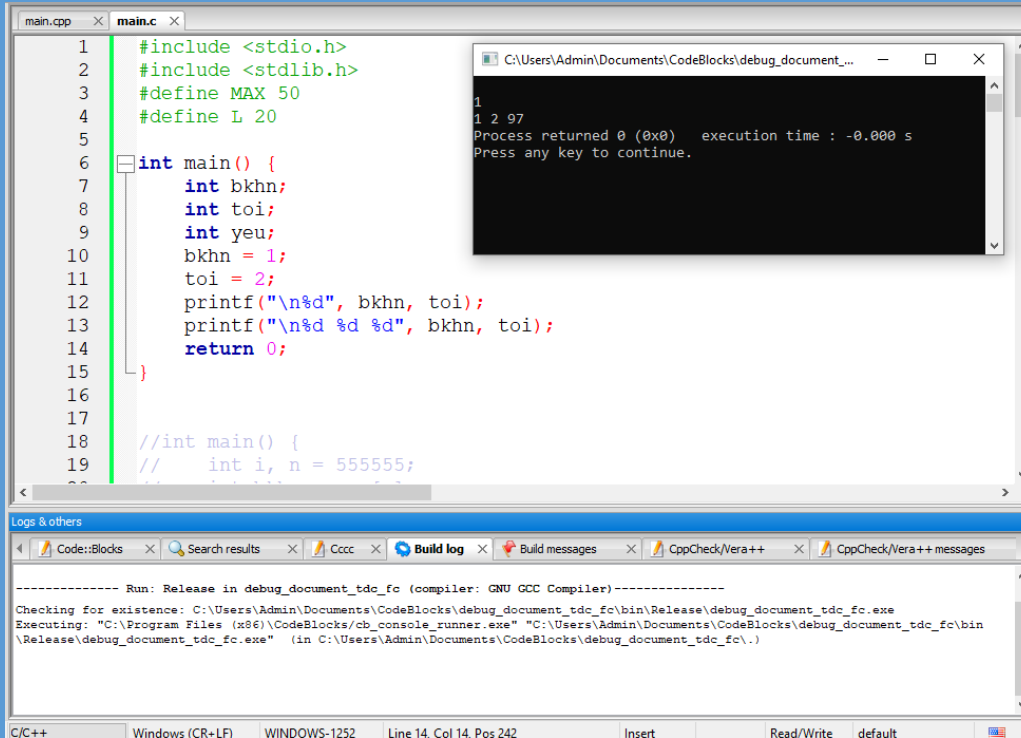
Cú pháp khai báo đúng của thư viện là:

#include <tên_thư_viện.h>

Nếu chương trình yêu cầu đưa ra độ chính xác là x chữ số thì cần chuyển định dạng là **%.xf** (tuân theo quy tắc làm tròn. Ở đây cần sửa lại thành **%.4f**)

#10 – LỖI THAO TÁC VỚI HÀM NHẬP XUẤT

LỖI XUẤT HIỆN



The screenshot shows the Code::Blocks IDE with a C++ file named `main.c`. The code defines `MAX` as 50 and `L` as 20. The `main` function declares `bkhn`, `toi`, and `yeu` as integers. It sets `bkhn = 1` and `toi = 2`, then prints them. The output window shows the values 1 and 2. The logs window shows the compilation and execution process.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #define MAX 50
4 #define L 20
5
6 int main() {
7     int bkhn;
8     int toi;
9     int yeu;
10    bkhn = 1;
11    toi = 2;
12    printf("\n%d", bkhn, toi);
13    printf("\n%d %d %d", bkhn, toi);
14    return 0;
15 }
16
17
18 //int main() {
19 //    int i, n = 555555;
```

Output window:

```
1
2 97
Process returned 0 (0x0)   execution time : -0.000 s
Press any key to continue.
```

Logs window:

```
----- Run: Release in debug_document_tdc_fc (compiler: GNU GCC Compiler)-----
Checking for existence: C:\Users\Admin\Documents\CodeBlocks\debug_document_tdc_fc\bin\Release\debug_document_tdc_fc.exe
Executing: "C:\Program Files (x86)\CodeBlocks\cb_console_runner.exe" "C:\Users\Admin\Documents\CodeBlocks\debug_document_tdc_fc\bin\Release\debug_document_tdc_fc.exe" (in C:\Users\Admin\Documents\CodeBlocks\debug_document_tdc_fc\..)
```

Lỗi logic

GIẢI THÍCH LỖI

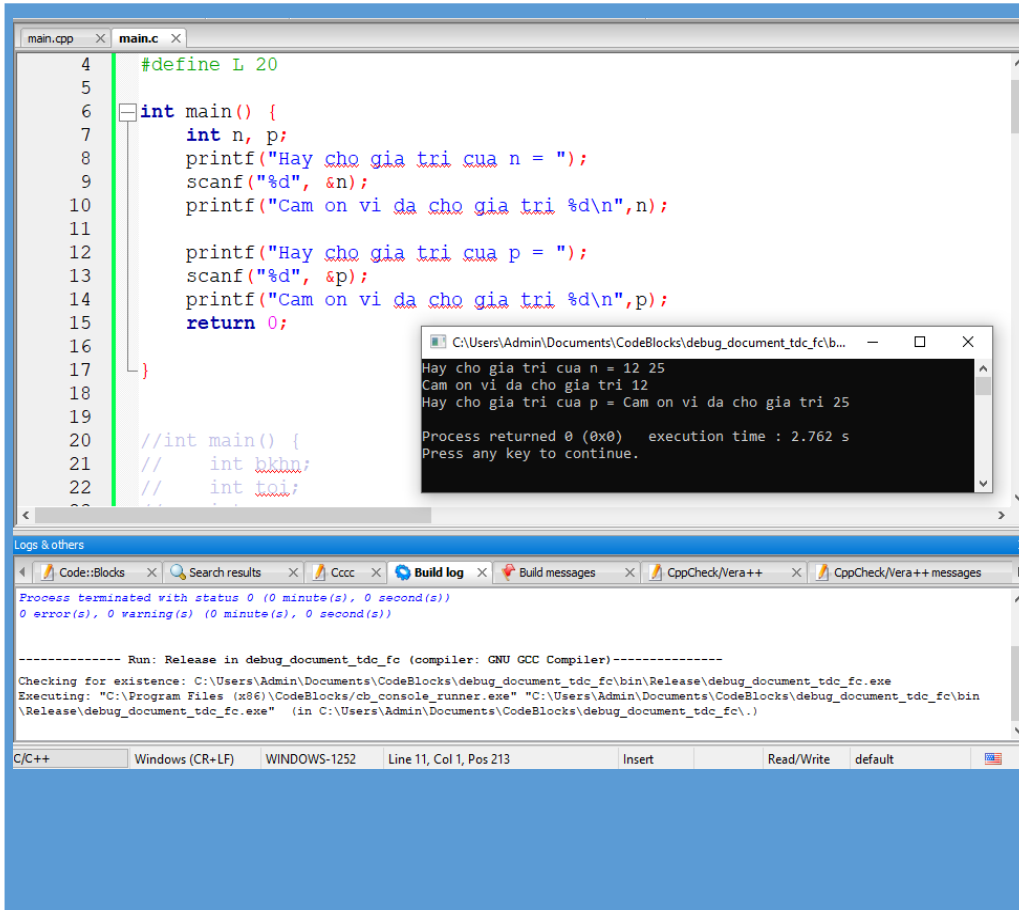
Khi thao tác với các hàm nhập xuất (ví dụ: `printf`, `scanf`), nếu cần đưa định dạng thì số lượng tham số phải tương ứng với số lượng định dạng.

CÁCH XỬ LÝ

Không có cách nào khác ngoài việc cẩn trọng khi viết chương trình

#11 – LỖI NHẦM DỮ LIỆU ĐỌC TỪ BÀN PHÍM

LỖI XUẤT HIỆN



```
main.cpp x main.c x
4 #define L 20
5
6 int main() {
7     int n, p;
8     printf("Hay cho gia tri cua n = ");
9     scanf("%d", &n);
10    printf("Cam on vi da cho gia tri %d\n", n);
11
12    printf("Hay cho gia tri cua p = ");
13    scanf("%d", &p);
14    printf("Cam on vi da cho gia tri %d\n", p);
15    return 0;
16
17 }
18
19 //int main() {
20 //    int bkhn;
21 //    int toi;
22 //    int...
```

Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))

----- Run: Release in debug_document_tdc_fc (compiler: GNU GCC Compiler)-----
Checking for existence: C:\Users\Admin\Documents\CodeBlocks\debug_document_tdc_fc\bin\Release\debug_document_tdc_fc.exe
Executing: "C:\Program Files (x86)\CodeBlocks\cb_console_runner.exe" "C:\Users\Admin\Documents\CodeBlocks\debug_document_tdc_fc\bin\Release\debug_document_tdc_fc.exe" (in C:\Users\Admin\Documents\CodeBlocks\debug_document_tdc_fc\.)

GIẢI THÍCH LỖI

Vấn đề nhầm lẫn dữ liệu khi đọc từ bàn phím.

Trong ví dụ, ta chưa kịp nhập dữ liệu cho p thì p đã nhận giá trị do việc ta sơ ý nhập thừa cho n.

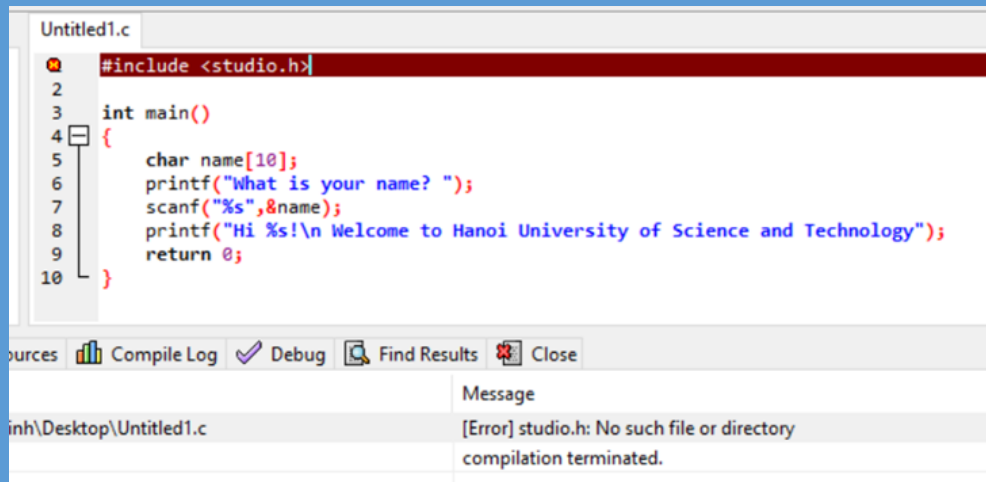
Vì dữ liệu ta gõ vào từ bàn phím được lưu ở bộ nhớ đệm truy xuất theo thứ tự của **Queue** – vào trước ra trước. Khi đó sau khi n được lấy ra, đến lệnh yêu cầu nhập p – dòng 13, chương trình thấy trong bộ nhớ đệm có rồi nên gán cho p luôn giá trị 25.

CÁCH XỬ LÝ

Ta sẽ gán giá trị đầu tiên đọc được cho biến và sẽ đọc nốt các ký tự còn lại trong bộ nhớ đệm bao gồm cả ký tự enter.

#12 – LỖI KHAI BÁO SAI TÊN THƯ VIỆN

LỖI XUẤT HIỆN



The screenshot shows a C program in a text editor. The code is as follows:

```
1 #include <studio.h>
2
3 int main()
4 {
5     char name[10];
6     printf("What is your name? ");
7     scanf("%s", &name);
8     printf("Hi %s!\n Welcome to Hanoi University of Science and Technology");
9     return 0;
10 }
```

Below the code editor, a message box displays the error:

Message
[Error] studio.h: No such file or directory
compilation terminated.

GIẢI THÍCH LỖI

Đây là lỗi về khai báo thư viện: Khai báo sai tên thư viện.

CÁCH XỬ LÝ

Sửa lại tên thư viện đúng: *stdio.h* – standard input output

#13 – LỖI SAI CÚ PHÁP

LỖI XUẤT HIỆN

GIẢI THÍCH LỖI

CÁCH XỬ LÝ

```
1  #include <stdio.h>
2
3  int main() {
4      int number;
5      printf("What is your ID ");
6      scanf("%d", number);
7      printf("Welcome to Hanoi University of Science and Technology!");
8      return 0;
9  }
```

Chương trình chỉ chạy đến hết hàm *scanf* → Hàm *scanf* chưa truyền địa chỉ của biến

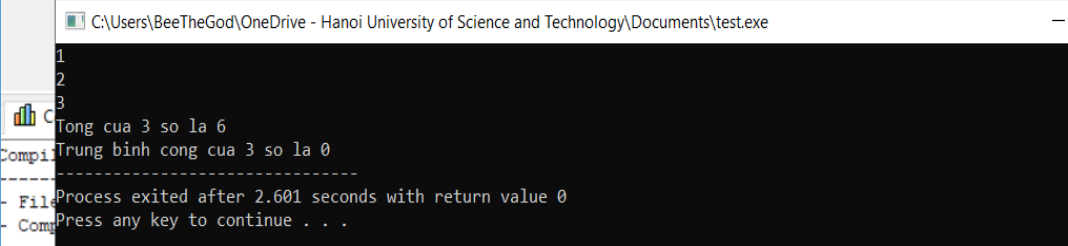
Xem lại dòng số 6, trước biến *number*, thêm kí tự '&'

```
What is your ID 65
-----
Process exited after 15.13 seconds with return value 3221225477
Press any key to continue . . .
```

#14 – LỖI SAI CÚ PHÁP

LỖI XUẤT HIỆN	GIẢI THÍCH LỖI	CÁCH XỬ LÝ
	Lỗi cú pháp: chuỗi in ra màn hình trong câu lệnh printf phải nằm trong dấu ngoặc kép (" ")	Xem lại dòng 8: sửa cặp dấu nháy đơn (' ') thành nháy kép.

#15 – LỖI SAI CHUỖI ĐỊNH DẠNG

LỖI XUẤT HIỆN	GIẢI THÍCH LỖI	CÁCH XỬ LÝ
<pre>1 #include <stdio.h> 2 int main() 3 { 4 int a,b,c; 5 scanf("%d",&a); 6 scanf("%d",&b); 7 scanf("%d",&c); 8 printf("Tong cua 3 so la %d\n",a+b+c); 9 printf("Trung binh cong cua 3 so la %d",(a+b+c)/3.0); 10 return 0; 11 }</pre> 	<p>Lỗi nằm ở dòng thứ 9, khi sinh viên nhầm chuỗi định dạng của float thành chuỗi định dạng của int.</p> <p>Lỗi này có thể xuất phát từ việc nhầm lẫn cơ bản giữa hai chuỗi định dạng.</p>	<p>Sinh viên cần sử dụng kiểu dữ liệu cũng như chuỗi định dạng sao cho chính xác.</p> <p>Với bài này, cần thay “%d” thành “%f” ở dòng 9.</p>

#16 – LỖI SAI CÚ PHÁP VÀ ĐỊNH DẠNG CỦA TOÁN TỬ

LỖI XUẤT HIỆN

Nhập vào 3 số thực a,b,c là 3 cạnh của 1 tam giác. Tính chu vi tam giác này? (Kiểm tra điều kiện đầu vào)

```
1  #include <stdio.h>
2
3  int main()
4  {
5      float a,b,c;
6      do{
7          scanf("%f%f%f", &a, &b, &c);
8      }while(a+b<=c & b+c<=a & c+a<=b);
9      printf("%f", a+b+c);
10 }
11
12 D:\Document HUST\2019-2\TA_TIN_DAI_CUONG\Code_Codeforces\D
13
14 1
15 2
16 C 10
17 13.000000
18
19 Total -----
```

GIẢI THÍCH LỖI

Chương trình sai về logic và cú pháp của toán tử. Một tam giác có 3 cạnh a,b,c thoả mãn:
 $a+b>c$, $c+a>b$, $b+c>a$.

Khi đó điều kiện vòng lặp để lặp lại cần:

Nếu $a+b \leq c$ HOẶC $c+a \leq b$ HOẶC $b+c \leq a$ thì các giá trị đầu vào là sai và cần nhập lại.

Phép toán logic HOẶC (OR) là `||`. Phép toán VÀ (AND) là `&&`. Còn `|` và `&` là phép toán OR và AND trong xử lý bit.

CÁCH XỬ LÝ

Sửa lại điều kiện vòng lặp thành:

$a+b \leq c$ ||

$b+c \leq a$ ||

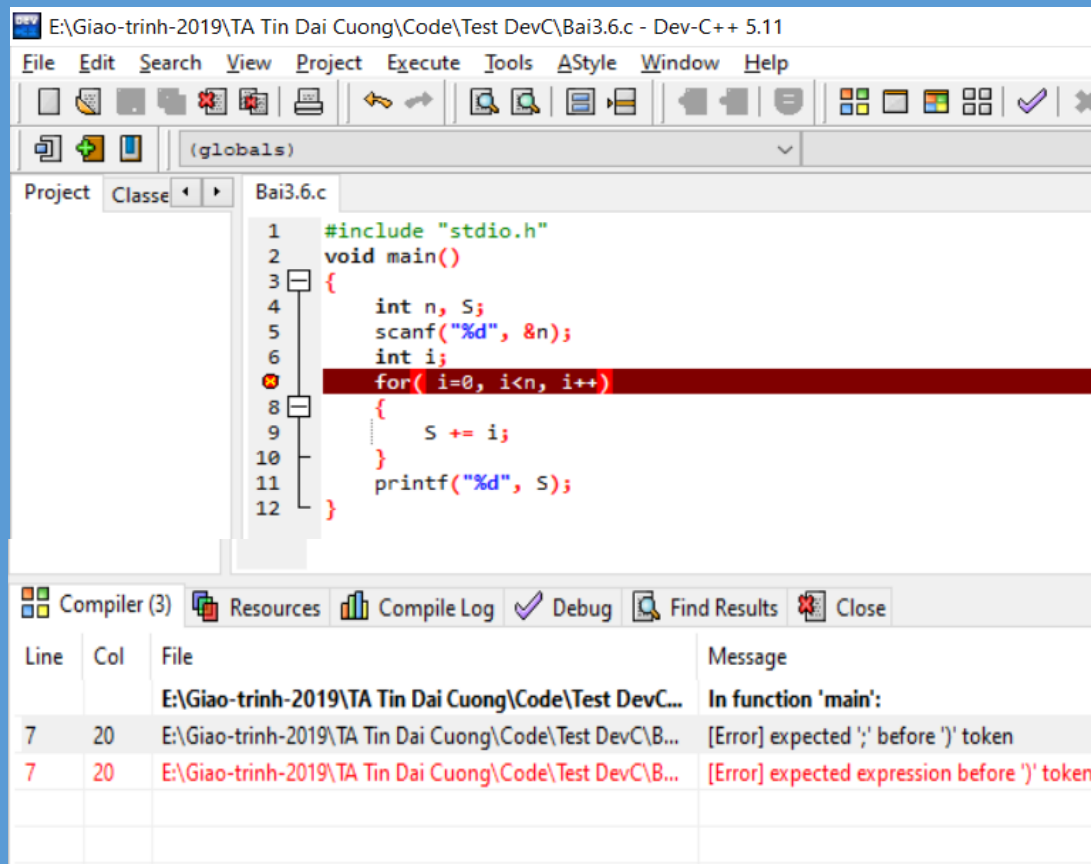
$c+a \leq b$

CẤU TRÚC ĐIỀU KHIỂN TRONG C



#17 – LỖI SAI CÚ PHÁP

LỖI XUẤT HIỆN



```
1 #include "stdio.h"
2 void main()
3 {
4     int n, S;
5     scanf("%d", &n);
6     int i;
7     for( i=0, i<n, i++)
8     {
9         S += i;
10    }
11    printf("%d", S);
12 }
```

Line	Col	File	Message
		E:\Giao-trinh-2019\TA Tin Dai Cuong\Code\Test DevC...	In function 'main':
7	20	E:\Giao-trinh-2019\TA Tin Dai Cuong\Code\Test DevC\B...	[Error] expected ';' before ')' token
7	20	E:\Giao-trinh-2019\TA Tin Dai Cuong\Code\Test DevC\B...	[Error] expected expression before ')' token

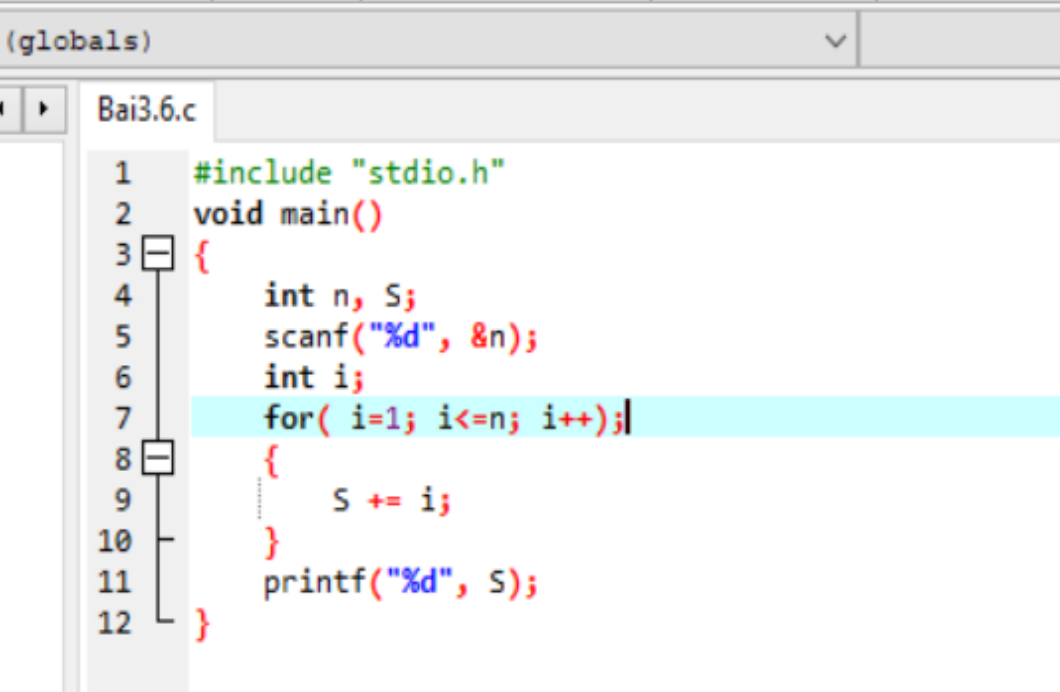
GIẢI THÍCH LỖI

Sai cú pháp vòng **for()**, báo lỗi thiếu dấu “;” do ngăn cách giữa các điều kiện trong vòng for phải bằng dấu “;”

CÁCH XỬ LÝ

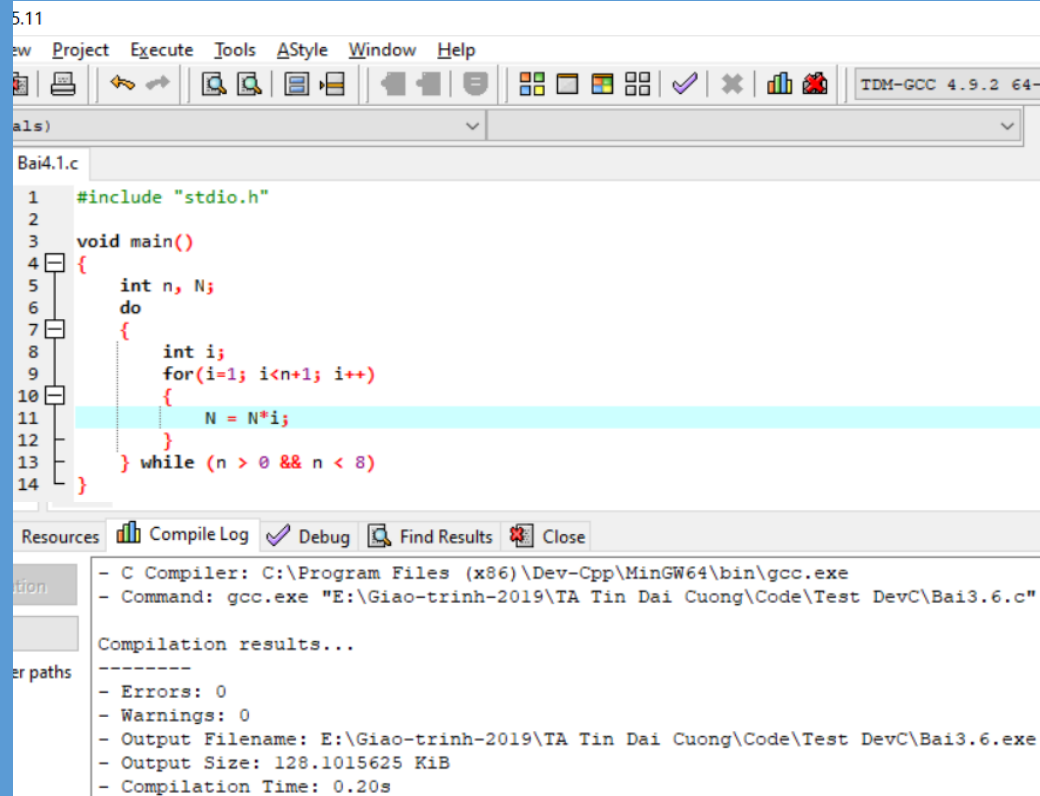
Thay dấu “,” thành “;”

#18 – LỖI SAI CÚ PHÁP

LỖI XUẤT HIỆN	GIẢI THÍCH LỖI	CÁCH XỬ LÝ
 <pre>(globals) Bai3.6.c 1 #include "stdio.h" 2 void main() 3 { 4 int n, S; 5 scanf("%d", &n); 6 int i; 7 for(i=1; i<=n; i++); 8 { 9 S += i; 10 } 11 printf("%d", S); 12 }</pre>	Không xuất hiện cảnh báo lỗi, nhưng chương trình không thực hiện đúng do sau vòng for có dấu “.”	Xóa dấu “.” sau for()

#19 – LỖI CHƯA KHỞI TẠO GIÁ TRỊ BAN ĐẦU

LỖI XUẤT HIỆN



```
5.11
new Project Execute Tools AStyle Window Help
als)
Bai4.1.c
1 #include "stdio.h"
2
3 void main()
4 {
5     int n, N;
6     do
7     {
8         int i;
9         for(i=1; i<n+1; i++)
10        {
11            N = N*i;
12        }
13    } while (n > 0 && n < 8)
14 }
```

Resources Compile Log Debug Find Results Close

- C Compiler: C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\gcc.exe
- Command: gcc.exe "E:\Giao-trinh-2019\TA Tin Dai Cuong\Code\Test DevC\Bai3.6.c"

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: E:\Giao-trinh-2019\TA Tin Dai Cuong\Code\Test DevC\Bai3.6.exe
- Output Size: 128.1015625 KiB
- Compilation Time: 0.20s

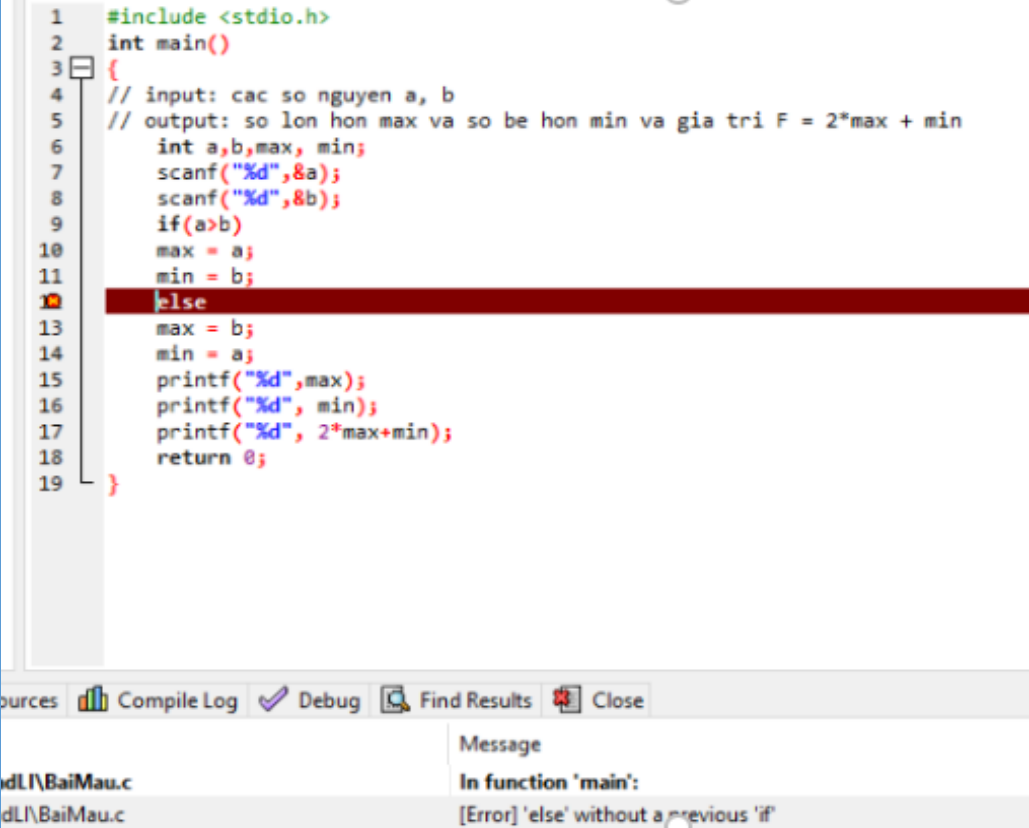
GIẢI THÍCH LỖI

Không xuất hiện lỗi, nhưng kết quả không đúng do chưa khởi tạo giá trị ban đầu cho N. Khi đó thực hiện câu lệnh $N = N \cdot i$ sẽ cho kết quả với giá trị mặc định của N.

CÁCH XỬ LÝ

Khai báo
 $\text{int } N = 1;$

#20 – LỖI SAI CÚ PHÁP

LỖI XUẤT HIỆN	GIẢI THÍCH LỖI	CÁCH XỬ LÝ
 <pre>1 #include <stdio.h> 2 int main() 3 { 4 // input: cac so nguyen a, b 5 // output: so lon hon max va so be hon min va gia tri F = 2*max + min 6 int a,b,max, min; 7 scanf("%d",&a); 8 scanf("%d",&b); 9 if(a>b) 10 max = a; 11 min = b; 12 else 13 max = b; 14 min = a; 15 printf("%d",max); 16 printf("%d", min); 17 printf("%d", 2*max+min); 18 return 0; 19 }</pre> <p>Message In function 'main': [Error] 'else' without a previous 'if'</p>	<p>Đây là lỗi về cú pháp if ... else ... khi có nhiều hơn một câu lệnh cần xử lý trong lệnh if, else</p>	<p>Cần đặt các câu lệnh sau khi kiểm tra điều kiện của if vào { }. Ngoài ra sau lệnh else cần thêm cặp { } để chương trình nhận biết đoạn gán min-max nếu điều kiện rẽ nhánh sai.</p>

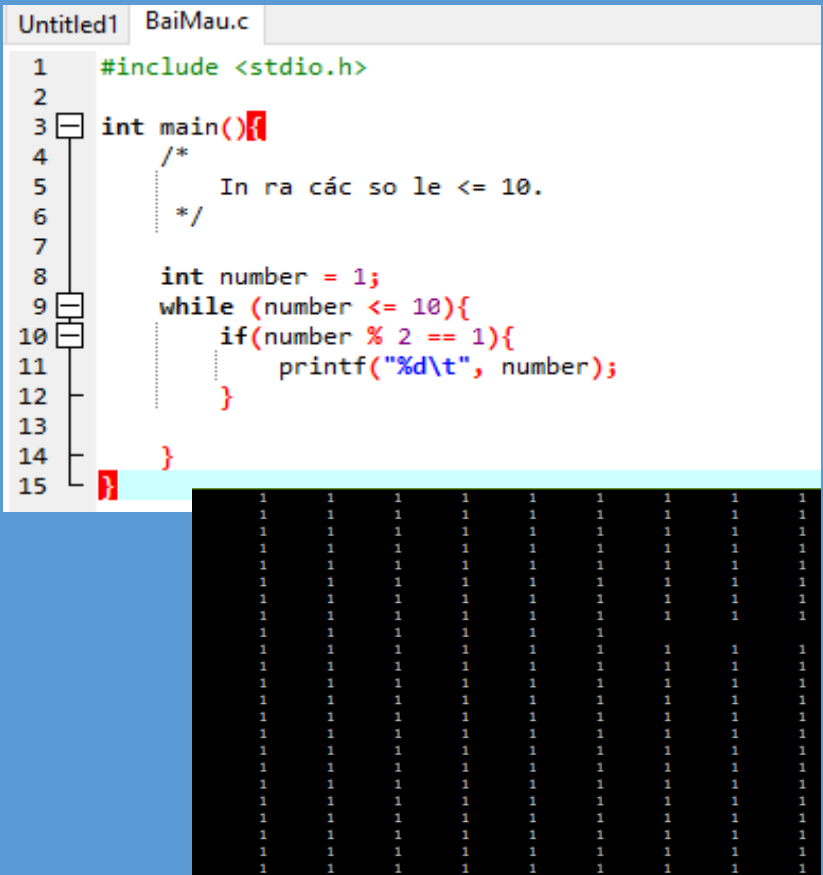
#21 – LỖI THIẾU BREAK

LỖI XUẤT HIỆN	GIẢI THÍCH LỖI	CÁCH XỬ LÝ
<pre>#include <stdio.h> int main() { // input: nhap phep toan +, -, *, / va 2 so nguyen tuong ung // output: Kq cua phep toan ap dung voi 2 so nguyen do int a, b; char opera; printf("\nNhap phep toan: "); scanf("%c", &opera); printf("\nNhap vao 2 so a, b: "); scanf("%d%d", &a, &b); switch (opera) { case '+': printf("%d + %d = %d", a, b, a + b); case '-': printf("%d - %d = %d", a, b, a - b); break; case '*': printf("%d * %d = %d", a, b, a * b); break; case '/': if(b == 0){ printf("Khong the chia cho 0!"); }else{ printf("%d / %d = %.2f", a, b, (float)a / b); } break; default: printf("Khong co phep toan %c!", opera); break; } }</pre> 	Lỗi do thiếu break trong case đầu tiên.	Bổ sung break vào case tương ứng

#22 – LỖI SAI CÚ PHÁP

LỖI XUẤT HIỆN	GIẢI THÍCH LỖI	CÁCH XỬ LÝ
	Lỗi này do chưa khai báo biến đếm i trong vòng for	Khai báo biến i

#23 – LỖI LẶP VÔ HẠN

LỖI XUẤT HIỆN	GIẢI THÍCH LỖI	CÁCH XỬ LÝ
 <pre> 1 #include <stdio.h> 2 3 int main() 4 /* 5 * In ra các so le <= 10. 6 */ 7 8 int number = 1; 9 while (number <= 10){ 10 if(number % 2 == 1){ 11 printf("%d\t", number); 12 } 13 } 14 15 </pre> <p>The output shows a continuous stream of '1' characters, indicating the loop never terminates.</p>	<p>Vòng lặp while không có điều kiện dừng khiến chương trình bị lặp vô hạn</p>	<p>Bổ sung điều kiện để vòng lặp while gặp điều kiện dừng.</p>

#24 – LỖI SAI CÚ PHÁP

LỖI XUẤT HIỆN

```
66 //Error with switch...case
67 void Function_3()
68 {
69     int a = 0;
70     switch(a)
71     {
72         case (a==0):
73             ///
74             printf("Function into case 1\n");
75             break;
76         case (a==1):
77             ///
78             break;
79         case (a==2):
80             ///
81             break;
82         case (a==3):
83             ///
84             break;
85         default:
86             ///
87             printf("Function into default, have not case is satisfy\n => Function was wrong!\n");
88     }
89 }
90
91
92
93
```

Compiler (5) Resources Compile Log Debug Find Results Close			
Line	Col	File	Message
		C:\Users\quang\Desktop\Nhap>List_Error.cpp	In function 'void Function_30':
72	9	C:\Users\quang\Desktop\Nhap>List_Error.cpp	[Error] 'a' cannot appear in a constant-expression
76	9	C:\Users\quang\Desktop\Nhap>List_Error.cpp	[Error] 'a' cannot appear in a constant-expression
79	9	C:\Users\quang\Desktop\Nhap>List_Error.cpp	[Error] 'a' cannot appear in a constant-expression
82	9	C:\Users\quang\Desktop\Nhap>List_Error.cpp	[Error] 'a' cannot appear in a constant-expression

GIẢI THÍCH LỖI

Lỗi với lệnh **switch...case**. Sử dụng sai cú pháp với điều kiện case.

CÁCH XỬ LÝ

Sinh viên thường nhầm lẫn và đưa vào case câu lệnh điều kiện như (a == 0). Nên nhớ từng case là từng trường hợp giá trị của biến điều kiện đưa vào lệnh switch. Ở ví dụ này là a. Câu lệnh phải viết lại như sau:

```
switch(a)
{
    case 0:
        ///
    case 1:
        ///
    .....
}
```

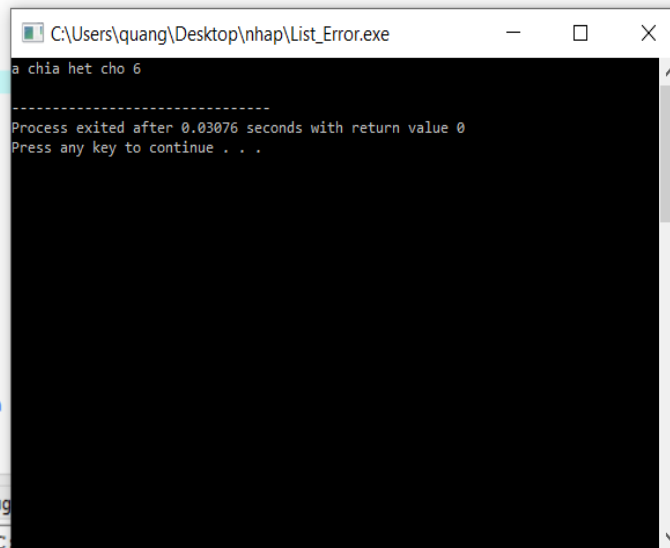
#25 – LỖI SAI CÚ PHÁP

LỖI XUẤT HIỆN	GIẢI THÍCH LỖI	CÁCH XỬ LÝ												
<div><pre>1 #include <stdio.h> 2 int main() 3 { 4 int var; 5 scanf("%d", &var); 6 printf("%s", var>10:"True":"False"); 7 } 8</pre><div>int __cdecl printf (const char * __restrict __Format, ...)</div><div>Compiler (2) Resources Compile Log Debug Find Results Close</div><table><thead><tr><th>Line</th><th>Col</th><th>File</th><th>Message</th></tr></thead><tbody><tr><td></td><td></td><td>D:\Documen...</td><td>In function 'main':</td></tr><tr><td>6</td><td>24</td><td>D:\Document ...</td><td>[Error] expected ')' before ':' token</td></tr></tbody></table></div>	Line	Col	File	Message			D:\Documen...	In function 'main':	6	24	D:\Document ...	[Error] expected ')' before ':' token	Sử dụng sai cú pháp của lệnh viết tắt if else	Sửa lại đúng cấu trúc: <điều kiện> ? <Đúng> : <Sai> Với bài này cần sửa thành: var>10?"True":"False"
Line	Col	File	Message											
		D:\Documen...	In function 'main':											
6	24	D:\Document ...	[Error] expected ')' before ':' token											

#26 – LỖI SAI ĐIỀU KIỆN

LỖI XUẤT HIỆN

```
108 //Error 30
109 //Error with if...else
110 void Function_5()
111 {
112     int a = 12;
113     if(a%6 == 0)
114     {
115         printf("a chia het cho 6\n");
116     }
117     else if(a%3 == 0)
118     {
119         printf("a chia het cho 3\n");
120     }
121     else if(a%2 == 0)
122     {
123         printf("a chia het cho 2\n");
124     }
125     else
126     {
127         printf("a khong chia het cho ca
128     }
129 }
```



GIẢI THÍCH LỖI

Ví dụ mắc lỗi logic trong khối lệnh if_else. Kiểm tra 1 số có chia hết 2, 3, 6 hay không. Tuy nhiên kết quả chỉ hiện thị chia hết cho 6.

CÁCH XỬ LÝ

Lưu ý khi thực hiện các khối lệnh If_else liên tiếp. Các điều kiện sau phải nằm ngoài hoàn toàn các khả năng có thể xảy ra ở trường hợp trước, tránh bị chồng chéo điều kiện logic dẫn đến sai kết quả.

#27 – LỖI SAI LOGIC

LỖI XUẤT HIỆN

Bài toán: Nhập vào số thực x, xuất kết quả ra mà hình giá trị hàm số sau:

$$f(x) = \begin{cases} \arccos(x) & , -1 \leq x \leq 1 \\ x + 5 & , x < -1 \cup x > 1 \end{cases}$$

```
DHBKHN.c
1  #include <stdio.h>
2  #include <math.h>
3  int main()
4  {
5      float x;
6      scanf("%f", &x);
7      if(-1<=x<=1) printf("%f", acos(x));
8      else printf("%f", x+5);
9  }
10
```

D:\Document HUST\2019-2\TA_TIN_DAI_CUONG\Code_Codeforces\DHBKHN.exe

```
-3
-1.#IND00
-----
Process exited after 3.227 seconds with return value 9
Press any key to continue . . .
```

GIẢI THÍCH LỖI

Đây là lỗi sai về logic.
Điều kiện trong lệnh if chưa chính xác, giả sử với đầu vào là -3 thì:
 $-1 \leq -3 \rightarrow \text{Sai} = 0$
 $0 \leq 1 \rightarrow \text{Đúng}$.
Do đó, biểu thức trong lệnh if sẽ được thực hiện với đầu vào này.

CÁCH XỬ LÝ

Sửa lại điều kiện là:
 $-1 \leq x \ \&\& \ x \leq 1$

#28 – LỖI LẶP VÔ HẠN

LỖI XUẤT HIỆN

Bài toán: Nhập vào số nguyên n, xuất ra màn hình chữ số lớn nhất của số đó.

```
DHBKHN.c
1  #include <stdio.h>
2
3  int main(){
4      int n, max = 0, donvi;
5      scanf("%d", &n);
6      while(n>0)
7          donvi = n%10;
8          if(donvi>max) max=donvi;
9          n /=10;
10         printf("%d", max);
11     }
```

D:\Document HUST\2019-2\TA_TIN_DAI_CUONG

456

GIẢI THÍCH LỖI

Chương trình lặp vô hạn, thiếu cặp {...} cho các lệnh bên trong vòng lặp while. Nếu không có {...} thì chương trình sẽ chỉ xét dòng đầu tiên ngay bên dưới while là thuộc vòng lặp, các dòng dưới nữa là ngoài vòng lặp.

CÁCH XỬ LÝ

Bổ sung cặp {...} bao bọc các dòng lệnh 7, 8 và 9.

#29 – LỖI SAI LOGIC

LỖI XUẤT HIỆN

Bài toán: Nhập vào số nguyên n, xuất ra màn hình các số từ 1 đến n mà không chia hết cho 5.

DHBKHN.c

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int n, i;
6      scanf("%d", &n);
7      for(i=1; i<=n; ++i){
8          if(i%5==0) break;
9          printf("%d ", i);
10     }
```

D:\Document HUST\2019-2\TA_TIN_DAI_CUONG\Code_Codeforces\DHBKHN.exe

```
12
1 2 3 4
-----
Process exited after 0.7573 seconds with return value 5
Press any key to continue . . .
```

GIẢI THÍCH LỖI

Lỗi sai logic. Khi chương trình gặp lệnh **break** sẽ nhảy ra khỏi vòng lặp đó .

CÁCH XỬ LÝ

Thay lệnh **break** bằng lệnh **continue**

#30 – LỖI KHAI BÁO BIẾN SAI

LỖI XUẤT HIỆN

Yêu cầu: Nhập vào số nguyên dương n (sử dụng do-while để kiểm tra). In ra màn hình các số nguyên dương nhỏ hơn n.

```
example.c
1  #include<stdio.h>
2
3  int main()
4  {
5      int n;
6      do
7      {
8          scanf("%d", &n);
9      }
10     while (n <= 0);
11
12     for (int i=1; i<n; i++)
13         printf("%d\n", i);
14
15     return 0;
16 }
```

Compiler (3) Resources Compile Log Debug Find Results Close

Line	Col	Message
		In function 'main':
12	2	[Error] 'for' loop initial declarations are only allowed in C99 or C11 mode
12	2	[Note] use option -std=c99, -std=gnu99, -std=c11 or -std=gnu11 to compile your code

GIẢI THÍCH LỖI

Đây là lỗi không cho phép khai báo biến trong vòng lặp **for**.

CÁCH XỬ LÝ

Khai báo biến **i** trước khi vào vòng lặp **for**.

#31 – LỖI SAI CÚ PHÁP

LỖI XUẤT HIỆN

Yêu cầu: Nhập vào số nguyên n khác 0 (sử dụng do-while để kiểm tra). In ra màn hình n.

```
example.c
1  #include<stdio.h>
2
3  int main()
4  {
5      int n;
6      do
7      {
8          scanf("%d", &n);
9      }
10     while !(n!=0);
11
12     printf("%d", n);
13
14     return 0;
15 }
```

Compiler (2) Resources Compile Log Debug Find Results Close

Line	Col	Message
10	8	In function 'main': [Error] expected '(' before '!' token

GIẢI THÍCH LỖI

Đây là lỗi thiếu dấu "(" trước "!".
Lưu ý cú pháp **do-while**:
do
{
 Câu lệnh;
}
while (biểu thức điều kiện);

CÁCH XỬ LÝ

Bổ sung thêm "(" và ")" đúng vị trí để sửa lỗi.

#32 – LỖI SAI LOGIC

LỖI XUẤT HIỆN

Yêu cầu: Nhập vào số nguyên dương $n < 20$ (sử dụng do-while để kiểm tra). In ra màn hình n.

example.c

```
1  #include<stdio.h>
2
3  int main()
4  {
5      int n;
6      do
7      {
8          scanf("%d", &n);
9      }
10     while (n > 0 && n < 20);
11     printf("%d", n);
12
13     return 0;
14 }
15
```

GIẢI THÍCH LỖI

Đây là lỗi sai về logic. Lưu ý với **do-while**: lệnh trong **do** sẽ tiếp tục thực hiện khi điều kiện trong **while** là đúng. Như vậy, câu lệnh trong **while** phải kiểm tra số nhập vào có nằm ngoài khoảng giá trị yêu cầu hay không.

CÁCH XỬ LÝ

Với đề bài $n < 20$ thì cần viết lại lệnh trong **while** như thế nào.

#33 – LỖI THIẾU LỆNH BREAK

LỖI XUẤT HIỆN

Yêu cầu: Nhập vào số 0 hoặc 1. Kiểm tra nếu số nhập vào là 0 thì thông báo “Ban da nhap so 0!”, ngược lại thông báo “Ban da nhap so 1!”.

```
example.c
1  #include<stdio.h>
2
3  int main()
4  {
5      int n;
6      do
7      {
8          scanf("%d", &n);
9      }
10     while (!(n==0||n==1));
11
12     switch(n)
13     {
14         case 0:
15             printf("Ban da nhap so 0!");
16         case 1:
17             printf("Ban da nhap so 1!");
18     }
19
20     return 0;
21 }
```

```
0
Ban da nhap so 0!Ban da nhap so 1!
-----
Process exited after 2.049 seconds with return value 0
Press any key to continue . . .
```

GIẢI THÍCH LỖI

Trình biên dịch không báo lỗi, tuy nhiên khi nhập vào 0 thì kết quả thu được lại là “Ban da nhap so 0!Ban da nhap so 1!”. Nguyên nhân khi đã in ra được “Ban da nhap so 0!” chưa sử dụng lệnh **break** để thoát ra khỏi câu lệnh **switch**.

CÁCH XỬ LÝ

Đặt thêm **break** vào đúng vị trí để thu được kết quả mong muốn.

#34 – LỖI THIẾU HOẶC THỪA DẤU NGOẶC

LỖI XUẤT HIỆN

```
15 //Forget the symbol.
16 void Function_1()
17 {
18     int i = 0;
19     int n = 100;
20     for(i = 0; i <n; i++)
21     {
22         for(int j = i; j <n; j++)
23         {
24             if(i%3 == j%3 && j%3 == 0)
25             {
26                 printf("%d\n", j);
27             }
28         }
29     }
30 }
```

Compiler (2)		Resources	Compile Log	Debug	Find Results	Close
Line	Col	File		Message		
30	1	C:\Users\quang\Desktop\nhap>List_Error.cpp		In function 'void Function_10':		
		C:\Users\quang\Desktop\nhap>List_Error.cpp		[Error] expected '}' at end of input		

GIẢI THÍCH LỖI

Lỗi thiếu hoặc thừa dấu đóng ngoặc của đoạn lệnh khối.

CÁCH XỬ LÝ

Khi thực hiện các câu lệnh khối trong C, sinh viên cần cẩn thận trong việc thêm các dấu đóng mở ngoặc mỗi khi thêm 1 câu lệnh khối (for, while, switch...case, if...else, do...while, hay một function...). Thực hiện “thụt” vào đầu dòng theo vị trí của từng khối một cách nhất quán giúp các dòng lệnh dễ nhìn, trình bày đẹp và nhất là dễ gỡ lỗi.

#35 – LỖI SAI CÚ PHÁP

LỖI XUẤT HIỆN

```
37 //Error with switch...case
38 void Function_2()
39 {
40     int a = 0;
41     switch(a)
42     {
43         case 0;
44             ///
45             printf("Function into case 1\n");
46             break;
47         case 1;
48             ///
49             break;
50         case 2;
51             ///
52             break;
53         case 3;
54             ///
55             break;
56         default;
57             ///
58             printf("Function into default, have not case is satisfy\n => Function wa
59     }
60 }
```

Compiler (6)			
Resources			
Compile Log			
Debug			
Find Results			
Close			
Line	Col	File	Message
		C:\Users\quang\Desktop\Nhap>List_Error.cpp	In function 'void Function_20':
43	9	C:\Users\quang\Desktop\Nhap>List_Error.cpp	[Error] expected ':' before ';' token
47	9	C:\Users\quang\Desktop\Nhap>List_Error.cpp	[Error] expected ':' before ';' token
50	9	C:\Users\quang\Desktop\Nhap>List_Error.cpp	[Error] expected ':' before ';' token
53	9	C:\Users\quang\Desktop\Nhap>List_Error.cpp	[Error] expected ':' before ';' token
56	10	C:\Users\quang\Desktop\Nhap>List_Error.cpp	[Error] expected ':' before ';' token

GIẢI THÍCH LỖI

Lỗi với lệnh **switch...case**. Không sử dụng dấu “:”, thay vào đó lại dùng dấu “;”

CÁCH XỬ LÝ

Trong C có khá ít lệnh sử dụng dấu “:”. Các lệnh thường phân tách nhau bởi dấu “;”. Do đó sinh viên mới làm quen với C thường nhầm lẫn ở loại lệnh này. Chú ý trong khối case không cần sử dụng dấu “{...}”. Tuy nhiên sử dụng cũng không sai.

MẢNG XÂU KÝ TỰ



#36 – LỖI KHỞI TẠO GIÁ TRỊ

LỖI XUẤT HIỆN

```
1  #include <stdio.h>
2
3  int main(){
4      char x[5] = "123456";
5
6      printf("%s", x);
7      return 0;
8  }
```

ces Compile Log Debug Find Results Close

daicuong\error.c

Message

In function 'main':

daicuong\error.c

[Warning] initializer-string for array of chars is too long [enabled by default]

GIẢI THÍCH LỖI

Đây là lỗi về khởi tạo giá trị cho mảng vượt quá số phần tử khai báo. Tuy mảng x khai báo có 5 phần tử, nhưng lưu ý chuỗi sẽ luôn kết thúc với ký tự '\0' cuối cùng. Với ví dụ minh họa thì chuỗi sẽ là "12345\0", nhưng ký tự '\0' sẽ không hiển thị ra mặc dù nó vẫn chiếm 1 phần tử.

CÁCH XỬ LÝ

Khai báo x[6], hoặc lớn hơn hoặc sử dụng khai báo sau: char x[]

#37 – LỖI LOGIC

LỖI XUẤT HIỆN

```
1  #include <stdio.h>
2
3  int main(){
4      int i;
5      int a[10] = {0,1,2,3,4,5,6,7,8,9};
6
7      // delete 1 value in array a
8      a[9] = '\0';
9
10     for(i=0; i < 10; i++){
11         printf("%d \n", a[i]);
12     }
13     return 0;
```

```
D:\code\c\tindaicuong\error.exe
0
1
2
3
4
5
6
7
8
9
0
-----
Process exited with return value 0
Press any key to continue . . .
```

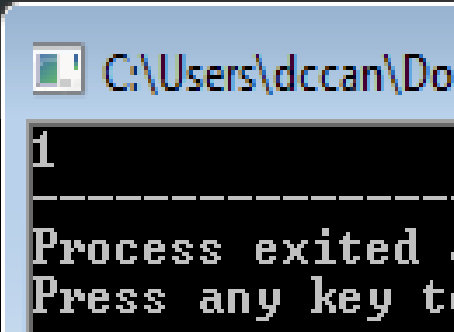
GIẢI THÍCH LỖI

Xóa 1 phần tử bất kỳ trong mảng, như hình bên là sai lỗi logic, đang gán giá trị cho phần tử cần xóa bằng 0

CÁCH XỬ LÝ

Cách giải quyết là tìm tới vị trí phần tử đó, tìm cách cho nó ra khỏi mảng và giảm kích thước mảng đi 1. Ví dụ: chuyển vị trí phần tử đó xuống cuối mảng rồi giảm kích thước mảng đi 1 phần tử là xong.

#38 – LỖI KHAI BÁO BIẾN SAI

LỖI XUẤT HIỆN	GIẢI THÍCH LỖI	CÁCH XỬ LÝ
<pre>int p=10 ; int mg[8]; for(i = 0;i<15;i++) { mg[i]=1 ; } printf("%d",p); return 0 ;</pre> 	Mảng khai báo với kích thước 8 phần tử nhưng sử dụng vượt quá số phần tử ban đầu khiến 1 số vùng nhớ khác bị ghi đè	Khai báo lại mảng mới hoặc Kiểm soát số lượng phần tử mảng khi viết code

#39 – LỖI SAI KHI SO SÁNH MẢNG

LỖI XUẤT HIỆN

```
1 #include<stdio.h>
2 #include<conio.h>
3 int i ;
4 int main() {
5     int mang1 [] = {1,2,3,4,5,6,7,8,9};
6     int mang2 [] = {1,2,3,4,5,6,7,8,9};
7     if(mang1!=mang2)
8         printf("hai mang khác nhau");
9     return 0 ;
10 }
```

Compile Log Debug Find Results X Close

Command: gcc.exe "C:\Users\dccan\Documents\TA\error.exe"

Compilation results...

Errors: 0
Warnings: 0
Output Filename: C:\Users\dccan\Documents\TA\error.exe
Output Size: 127.94921875 KiB
Compilation Time: 1.75s

C:\Users\dccan\Documents\TA\error.exe

hai mang khác nhau

Process exited after 0.04163 seconds
Press any key to continue . . .

GIẢI THÍCH LỖI

Đây là lỗi hay gặp trong C không thể so sánh 2 mảng bằng cách so sánh tên mảng, Vì tên mảng chỉ chứa địa chỉ của phần tử đầu tiên trong mảng.

CÁCH XỬ LÝ

Một cách xử lý hay được dùng là sử dụng vòng lặp duyệt từng phần tử trong 2 mảng rồi so sánh chúng với nhau.

#40 – LỖI SAI CÚ PHÁP

LỖI XUẤT HIỆN

```
function.c
1  #include<stdio.h>
2  int main(){
3      int i, n = 5;
4      int mang = {1,2,3,4,5};
5
6      for(i=0; i<n; i++){
7          printf("%d \n", mang[i]);
8      }
9      return 0;
10 }
```

Compiler (10) Resources Compile Log Debug Find Results Close

Line	Col	File	Message
4	2	E:\Tai lieu tro giang\function.c	In function 'main':
4	2	E:\Tai lieu tro giang\function.c	[Warning] excess elements in scalar initializer
4	2	E:\Tai lieu tro giang\function.c	[Warning] (near initialization for 'mang')
4	2	E:\Tai lieu tro giang\function.c	[Warning] excess elements in scalar initializer
4	2	E:\Tai lieu tro giang\function.c	[Warning] (near initialization for 'mang')
4	2	E:\Tai lieu tro giang\function.c	[Warning] excess elements in scalar initializer
4	2	E:\Tai lieu tro giang\function.c	[Warning] (near initialization for 'mang')
4	2	E:\Tai lieu tro giang\function.c	[Warning] excess elements in scalar initializer
4	2	E:\Tai lieu tro giang\function.c	[Warning] (near initialization for 'mang')
7	23	E:\Tai lieu tro giang\function.c	[Error] subscripted value is neither array nor pointer nor vector

GIẢI THÍCH LỖI

Chương trình không chạy được và báo lỗi ở dòng 7.
+Nguyên nhân:
Tại dòng 4, khi khai báo biến mảng với tên 'mang' người lập trình quên cặp ký tự '[]'. Đây là lỗi cú pháp khai báo biến mảng.

CÁCH XỬ LÝ

Xem lại và sửa phần khai báo biến mảng. Lưu ý khi khai báo biến mà không kèm giá trị khởi tạo thì phải nhớ xác định số lượng phần tử tối đa của mảng. Ví dụ: **int array[100];**. Còn nếu muốn khai báo biến kèm giá trị khởi tạo thì có thể không cần xác định số lượng phần tử tối đa. Ví dụ: **int array[] = {1,2,3,5};**

#41 – LỖI KHI SỬ DỤNG MẢNG

LỖI XUẤT HIỆN

```
int main() {  
    int i;  
    int array[5] = {0, 1, 2, 3, 4};  
  
    for (i = 0; i < 10; i++) {  
        printf("array[%d] - %d - %d\n", i, array[i], &array[i]);  
    }  
    return 0;  
}
```

	Gia tri	Dia chi
array[0]	0	6487552
array[1]	1	6487556
array[2]	2	6487560
array[3]	3	6487564
array[4]	4	6487568
array[5]	0	6487572
array[6]	1	6487576
array[7]	7	6487580
array[8]	7410592	6487584
array[9]	0	6487588

GIẢI THÍCH LỖI

Mặc dù khai báo mảng array với kích cỡ 5 phần tử, nhưng khi sử dụng array[5], array[6]... vẫn không báo lỗi. Do các địa chỉ array[5], array[6]... vẫn tồn tại tiếp nối sau array[4], và chứa các giá trị “rác”.

CÁCH XỬ LÝ

Chú ý khi khai báo kích cỡ mảng. Chú ý chỉ mục index khi sử dụng phần tử mảng.

#42 – LỖI VƯỢT QUÁ KÍCH THƯỚC MẢNG

LỖI XUẤT HIỆN	GIẢI THÍCH LỖI	CÁCH XỬ LÝ
<pre>int main() { int i, n = 10; int array[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}; int value = 25, position = 5; // Hàm thêm phần tử giá trị value vào vị trí position addElement(array, n, value, position); for (i = 0; i < n; i++) { printf("array[%d] = %d\n", i, array[i]); } return 0; }</pre> <pre>array[0] = 0 array[1] = 1 array[2] = 2 array[3] = 3 array[4] = 4 array[5] = 25 array[6] = 5 array[7] = 6 array[8] = 7 array[9] = 8</pre>	<p>Bài toán thêm phần tử vào mảng thường bị lỗi vượt quá kích cỡ mảng. Như ví dụ thì mảng bị tràn ra ngoài nên mất giá trị 9.</p>	<p>Khai báo mảng với kích cỡ lớn. Hoặc kiểm tra mảng đã đầy chưa, mỗi lần thêm phần tử thì phải nói rộng kích cỡ mảng.</p>

#43 – LỖI NHẦM HÀNG VÀ CỘT

LỖI XUẤT HIỆN

```
int main() {
    int i, j, m = 2, n = 4;
    int array[m][n]; // Ma tran 2 hang 4 cot
    printf("\nNhap vao ma tran:\n");
    for (i = 0; i < m; i++) {
        for (j = 0; j < n; j++) {
            printf("Phan tu hang %d cot %d: array[%d][%d] = ", i+1, j+1, i+1, j+1);
            scanf("%d", &array[i][j]);
        }
    }
    printf("\nMa tran vua nhap:\n");
    for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++) {
            printf("%d\t", array[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

```
Nhap vao ma tran:
Phan tu hang 1 cot 1: array[1][1] = 11
Phan tu hang 1 cot 2: array[1][2] = 12
Phan tu hang 1 cot 3: array[1][3] = 13
Phan tu hang 1 cot 4: array[1][4] = 14
Phan tu hang 2 cot 1: array[2][1] = 21
Phan tu hang 2 cot 2: array[2][2] = 22
Phan tu hang 2 cot 3: array[2][3] = 23
Phan tu hang 2 cot 4: array[2][4] = 24
```

```
Ma tran vua nhap:
11          12
21          22
7407448    0
4           0
```

GIẢI THÍCH LỖI

Mảng nhiều chiều - ở đây là mảng hai chiều. Khai báo ma trận 2 hàng 4 cột nhưng in ra ma trận lại là 4 hàng 2 cột. Ma trận in ra bị mất một số giá trị và lại có thêm một số giá trị lạ.

(Lệnh in ma trận bị nhầm: i chạy từ 0 đến n và j chạy từ 0 đến m).

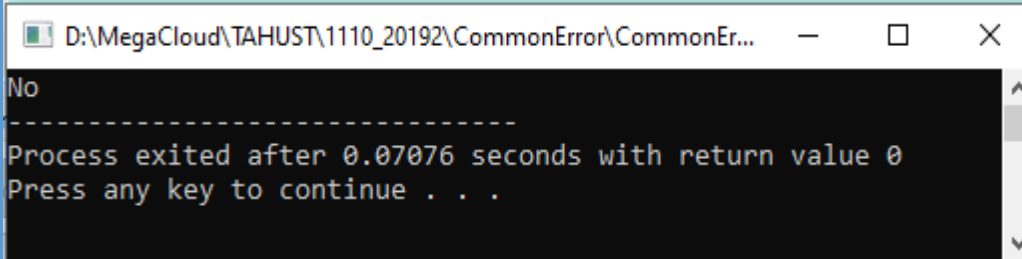
CÁCH XỬ LÝ

Chú ý khi lập trình không để nhầm lẫn kích cỡ từng chiều của mảng.

#44 – LỖI SO SÁNH MẢNG

LỖI XUẤT HIỆN

```
int main(int argc, char *argv[]) {  
    char str1[50] = "Tin hoc dai cuong hoc online vi Covid 19";  
    char str2[50] = "Tin hoc dai cuong hoc online vi Covid 19";  
    if (str1 == str2){  
        printf("Yes");  
    }else{  
        printf("No");  
    }  
    return 0;  
}
```



No

Process exited after 0.07076 seconds with return value 0
Press any key to continue . . .

GIẢI THÍCH LỖI

So sánh 2 xâu với nhau. Lỗi xảy ra khi sử dụng `==`. Mặc dù chương trình có thể compile được nhưng chương trình sẽ in sai kết quả. Cụ thể ở đây, mong muốn in ra **Yes** nhưng ở chương trình lại in ra **No**.

Lý do lỗi: chương trình đang so sánh 2 địa chỉ của 2 mảng `str1` và `str2`

CÁCH XỬ LÝ

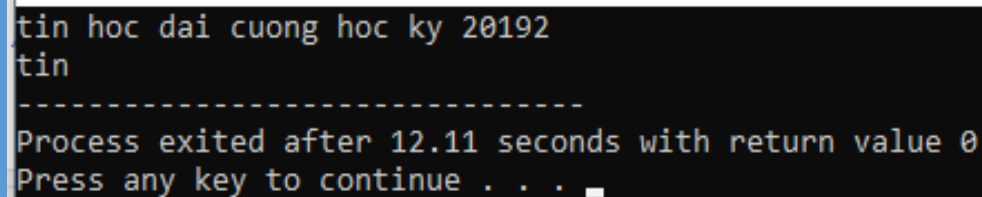
Khi so sánh 2 xâu sử dụng hàm **`strcmp()`** để so sánh

#45 – LỖI SỬ DỤNG HÀM KHÔNG PHÙ HỢP

LỖI XUẤT HIỆN

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

int main(int argc, char *argv[]) {
    char str1[50];
    scanf("%s", str1);
    printf("%s", str1);
    return 0;
}
```



GIẢI THÍCH LỖI

Nhập vào sử dụng `scanf`. Khi in ra màn hình chỉ nhận một phần vào nhập vào. Cụ thể là phần trước dấu Space (“tin”) thay vì toàn bộ vào (“tin hoc dai cuong hoc ky 20192”).

Lý do lỗi: hàm `scanf()` sẽ dừng tổng hợp ký tự khi gặp dấu xuống dòng, dấu cách.

CÁCH XỬ LÝ

Sử dụng hàm `gets()`. Tuy nhiên hàm `gets()` này đã bị xóa trong phiên bản C11. Gợi ý sử dụng hàm `fgets()` thay thế.

#46 – LỖI THIẾU LỆNH **FFLUSH**

LỖI XUẤT HIỆN

Untitled1.c

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int c;
6      char s[100];
7      printf("c = "); scanf("%d", &c);
8      printf("s = ");
9      fgets(s, 100, stdin);
10     printf("c = %d\n", c);
11     printf("s = %s\n", s);
12     return 0;
13 }
```

C:\Users\Linh\Desktop\Untitled1.exe

```
C:\Users\Linh\Desktop\Untitled1.exe
c = 5
s = c = 5
s =
```

```
-----
Process exited after 2.747 seconds with return value 0
Press any key to continue . . .
```

GIẢI THÍCH LỖI

Xâu s không được nhập vào. Do hàm **scanf** kết thúc bằng kí tự `\n`, kí tự này được lưu trong bộ đệm. Lệnh **fgets()** khi chạy sẽ nhận kí tự này từ bộ đệm và kết thúc nhập.

CÁCH XỬ LÝ

Xóa bộ đệm trước khi nhập xâu. Sử dụng lệnh **fflush()**.

Thêm lệnh **fflush(stdin)** trước lệnh **fgets()**

#47 – LỖI SAI THỨ TỰ THAM SỐ

LỖI XUẤT HIỆN

BaiMau.c

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdio.h>
4  int main()
5  /*
6      copy mot chuoi sang mang khac
7  */
8  char A[255], B[255];
9  printf("Nhap chuoi: ");
10 gets(A);
11 strcpy(A, B);
12 printf("Chuoi dich: ");
13 puts(B);
14 getch();
15 return 0;
16
```

E:\DownloadLI\BaiMau.exe

Nhap chuoi: Tin hoc
Chuoi dich: d

GIẢI THÍCH LỖI

Sai thứ tự tham số khi sử dụng hàm **strcpy()**

CÁCH XỬ LÝ

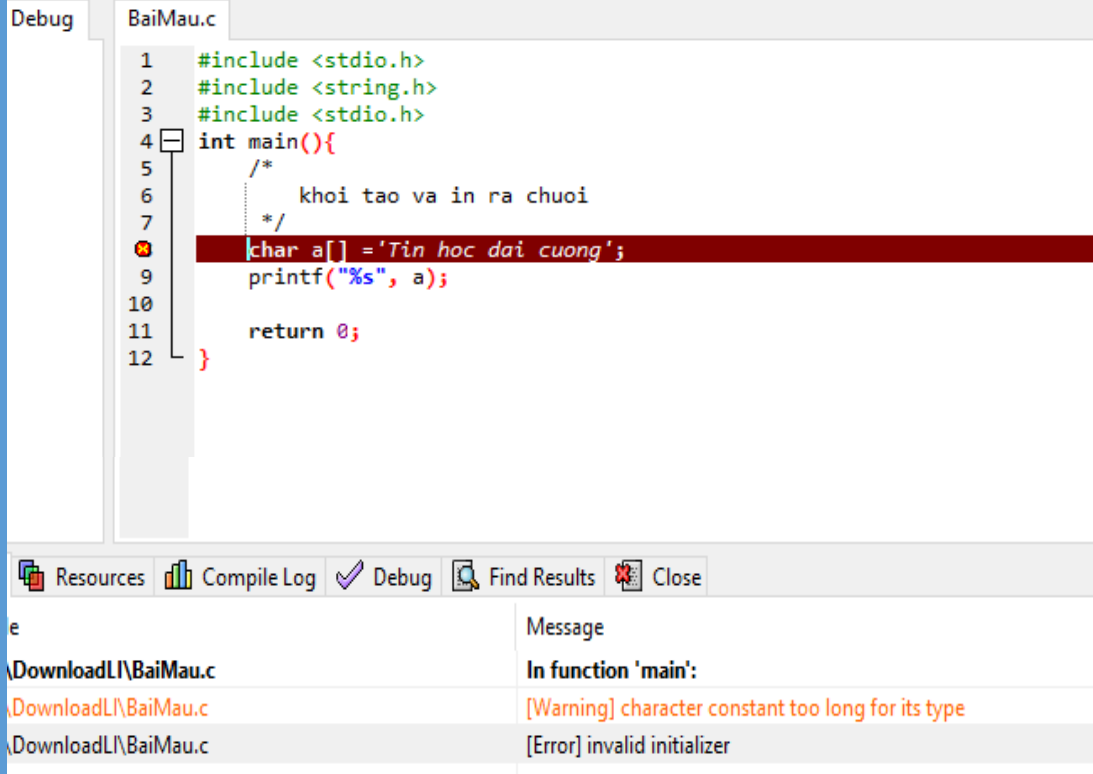
Kiểm tra lại thứ tự chuỗi nguồn, chuỗi đích khi sử dụng hàm **strcpy()**

#48 – LỖI NHẦM NGOẶC ĐƠN VÀ NGOẶC KÉP

LỖI XUẤT HIỆN

GIẢI THÍCH LỖI

CÁCH XỬ LÝ



```
1 #include <stdio.h>
2 #include <string.h>
3 #include <stdio.h>
4 int main(){
5     /*
6      *   khởi tạo và in ra chuỗi
7      */
8     char a[] = 'Tin học đại cương';
9     printf("%s", a);
10
11     return 0;
12 }
```

Resources Compile Log Debug Find Results Close

Message

In function 'main':

[Warning] character constant too long for its type

[Error] invalid initializer

Lỗi do sử dụng sai ‘ ‘ và “ ”. Lưu ý ‘ ‘ dùng cho ký tự còn “ ” dùng cho chuỗi.

Sửa lại cách biểu diễn chuỗi.

#49 – LỖI LẬP TRÌNH CHƯA ĐÚNG

LỖI XUẤT HIỆN

BaiMau.c

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdio.h>
4
5  int main()
6  /*
7   *   khai tạo và in ra chuỗi
8   */
9   int i, n;
10  scanf("%d", &n);
11  char names[n][50];
12  for(i = 0; i < n; i++){
13      fflush(stdin); gets(names[i]);
14  }
15  printf("Danh sach hoc sinh :%s \n ", names);
16  return 0;
17
```

```
3
Nguyen van a
Le van b
Hoang van c
Danh sach hoc sinh :Nguyen van a
-----
Process exited after 12.33 seconds with return value 0
Press any key to continue . . .
```

GIẢI THÍCH LỖI

Lỗi do names là một mảng nhiều chuỗi nên dùng **printf** với **%s** sẽ chỉ in ra chuỗi đầu tiên trong mảng thay vì tất cả các chuỗi trong mảng.

CÁCH XỬ LÝ

Tìm hiểu thêm phương pháp in danh sách các chuỗi (gợi ý dùng vòng **for**)

#50 – LỖI SAI CÚ PHÁP

LỖI XUẤT HIỆN

GIẢI THÍCH LỖI

CÁCH XỬ LÝ

The screenshot shows a code editor window titled 'main.cpp' with the following code:

```
1
2
3
4 #include <stdio.h>
5
6 int main() {
7     char name[10];
8     printf("what is your name? : ");
9     scanf("%c", name);
10    printf("Hi! %s \n Welcome to Hanoi University of Science and Technology", name);
11    return 0;
12 }
```

Below the code editor, a terminal window titled 'C:\Users\Admin\Documents\CodeBlocks\test_trunglq\bin\Release\test_trunglq.exe' shows the program's output:

```
what is your name? : Linh
Hi! L
Welcome to Hanoi University of Science and Technology
Process returned 0 (0x0)   execution time : 14.621 s
Press any key to continue.
```

At the bottom, the 'Logs & others' panel shows the build log:

```
mingw32-g++.exe -Wall -fexceptions -O2 -c C:\Users\Admin\Documents\CodeBlocks\test_trunglq\main.cpp -o obj\Release\main.o
mingw32-g++.exe -o bin\Release\test_trunglq.exe obj\Release\main.o -s
Output file is bin\Release\test_trunglq.exe with size 8.50 KB
Process terminated with status 0 (0 minute(s), 0 second(s))
0 error(s), 0 warning(s) (0 minute(s), 0 second(s))
```

The status bar at the bottom indicates 'Line 15, Col 1, Pos 232'.

Nhập xâu sử dụng **%c**, biến sẽ chỉ lưu trữ 1 ký tự đầu tiên được nhập vào

Xem lại dòng 9, sửa **%c** thành **%s** (dùng để đọc vào chuỗi ký tự không có khoảng trắng)

