

HƯỚNG DẪN CÁCH SỬA LỖI CHƯƠNG TRÌNH C/C++

LỜI NÓI ĐẦU

Qua kinh nghiệm nhiều năm giảng dạy tôi nhận thấy nhiều bạn SV gặp khó khăn khi học lập trình. Trong đó có một nguyên nhân lớn là các bạn không biết cách sửa lỗi một chương trình.

Khi chương trình có lỗi các bạn thường bỏ qua không đọc các câu thông báo lỗi (trong trường hợp lỗi cú pháp, lỗi thời gian chạy) hay việc sử dụng công cụ Debug (trong trường hợp lỗi logic). Chính vì thế các bạn mất rất nhiều thời gian để loay hoay đọc từng dòng code hay so sánh xem có giống với tài liệu hay thậm chí là “bạn kể bên” không ?

Từ việc mất thời gian quá lâu để sửa lỗi mà vẫn không xong khiến nhiều bạn chán nản rồi dẫn tới “mất căn bản” không theo nổi môn học.

Thực ra nếu chúng ta nắm vững được kỹ thuật sửa lỗi thì việc sửa lỗi của chương trình không khó mà nhiều khi lại làm chúng ta hứng thú khi khám phá được những “lỗi hay” hay những lưu ý quan trọng bổ sung cho kỹ năng lập trình của bản thân.

Tài liệu này được viết để giúp các bạn SV hiểu được các dạng lỗi và cách sửa lỗi. Các bạn cũng có thể dùng tài liệu này để tra cứu các câu thông báo lỗi thường gặp khi lập trình.

PHẦN 1

TỔNG QUAN VỀ CÁC LOẠI LỖI

Có 3 loại lỗi đối với một chương trình:

1. Lỗi cú pháp (Syntax Errors):

- Mỗi ngôn ngữ lập trình đều có một bộ quy tắc về ngữ pháp riêng mà người lập trình phải tuân thủ. Lỗi cú pháp xảy ra khi mã nguồn (source code) không tuân thủ theo quy tắc này.
- Trình biên dịch sẽ phát hiện ra lỗi này tại thời điểm biên dịch chương trình và hiển thị các lỗi này trong danh sách lỗi (Error list)

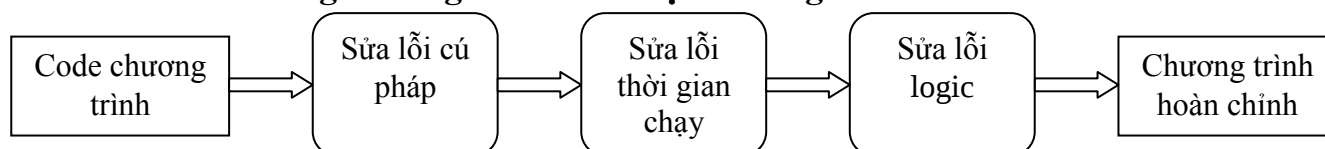
2. Lỗi thời gian chạy (Runtime Errors):

- Các lỗi xảy ra trong quá trình thực thi chương trình được gọi là lỗi thời gian chạy.
- Khi một lỗi thời gian chạy xảy ra, máy tính dừng việc thực hiện chương trình và hiển thị một câu thông báo lỗi.

3. Lỗi logic (Logical Errors):

- Các lỗi trong logic của chương trình được gọi là lỗi logic.
- Trình biên dịch không thể phát hiện các lỗi logic. Một chương trình có các lỗi logic được biên dịch (dịch) và chạy thành công, nhưng nó không cho kết quả chính xác.
- Khi chương trình cho ra kết quả sai hoặc không cho ra kết quả, sau khi xem lại code mà vẫn không tìm được nguyên nhân, chúng ta nên dùng công cụ Debug để kiểm tra cách thức thi hành của chương trình hoặc xem kết quả chạy chương trình tại những thời điểm (dòng lệnh) cần thiết.

4. Các bước thông thường để sửa lỗi một chương trình C/C++:



PHẦN 2

CÁCH SỬA CÁC LOẠI LỖI

I. CÁCH SỬA LỖI CÚ PHÁP

1. Cách xem lỗi cú pháp

- Để biên dịch chương trình tìm lỗi cú pháp: Chọn view/compile (Ctrl+F7) hay thực thi luôn chương trình Start Debugging (F5), khi đó nếu chương trình có lỗi cú pháp thì danh sách lỗi sẽ được thông báo.

- Để chọn chế độ hiển thị danh sách lỗi: Chọn View/Error list

2. Cách sửa lỗi cú pháp

- Nhấp đúp vào mỗi dòng thông báo lỗi, chương trình dịch sẽ chỉ ra dòng bị lỗi tương ứng trong chương trình. Ta cần đọc hiểu câu thông báo lỗi để sửa lỗi nhanh mà không phải “đoán mò” rất mất thời gian.

- Lưu ý:

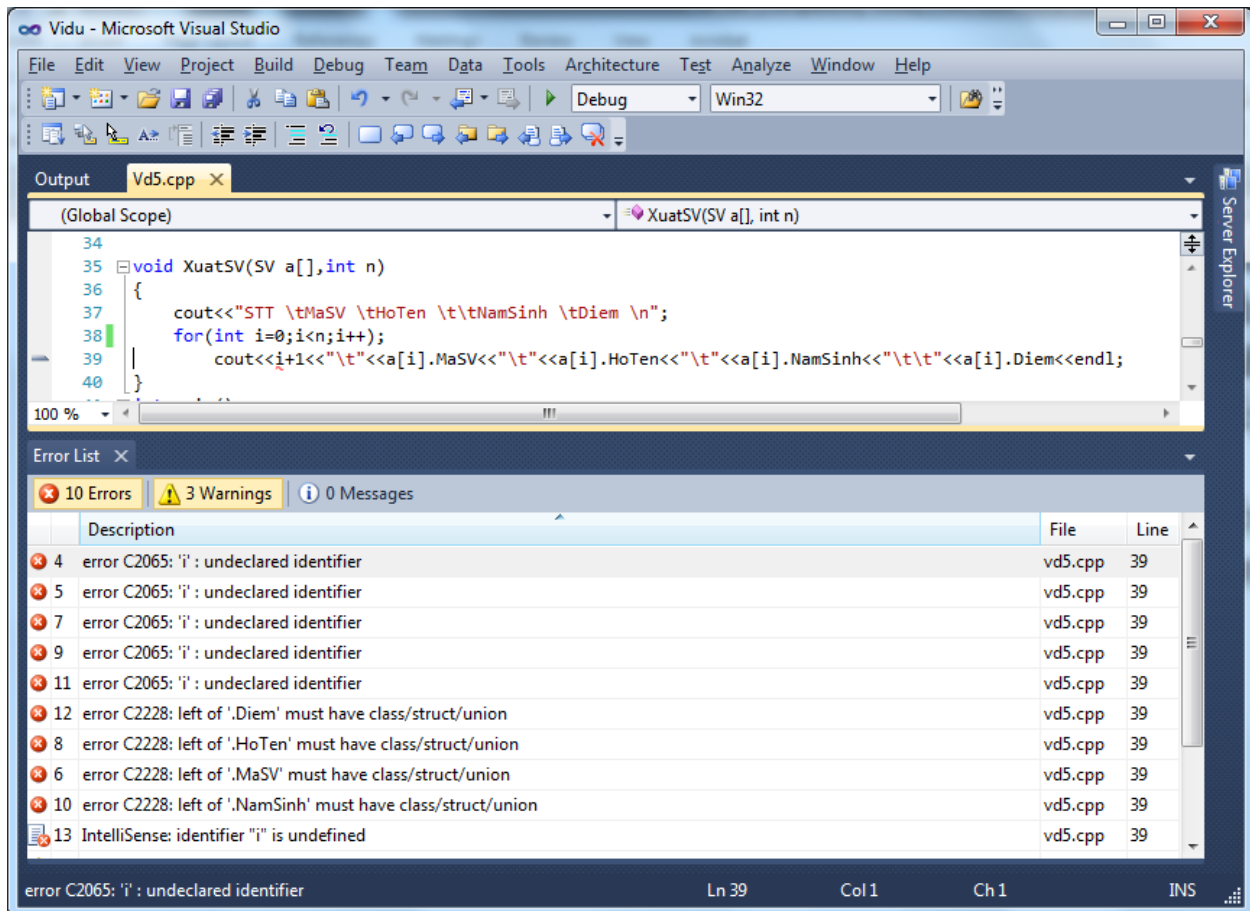
+ Cần chú ý là warning (cảnh báo) không phải là lỗi.

+ Trong danh sách lỗi, chỉ có lỗi đầu tiên (lỗi được đánh số nhỏ nhất và không phải lúc nào cũng được đánh số từ 1) là được thông báo chính xác, các lỗi tiếp theo sau có thể là lỗi phát sinh do lỗi đầu tiên. Vì vậy để tránh mất thời gian ta **chỉ sửa lỗi đầu tiên** sau đó dịch lại chương trình để tìm lỗi tiếp theo.

Trong hình ở trang sau, chương trình có 10 lỗi cú pháp và lỗi đầu tiên được đánh số 4 (vì trước đó có 3 cảnh báo). Sau khi ta sửa được lỗi đầu tiên, các lỗi khác sẽ “biến mất” sau khi biên dịch lại chương trình.

+ Khi chương trình dịch chạy đến dòng trong thông báo lỗi thì phát hiện ra lỗi nên có khả năng lỗi không nằm trên dòng này mà nằm trên dòng trước đó.

Trong hình ở trang sau, trình biên dịch phát hiện lỗi ở dòng 39 nhưng lỗi lại nằm ở dòng 38 (do dư dấu ‘;’ ở cuối dòng 38)



3. Các câu thông báo lỗi cú pháp thường gặp

Do tâm lý ngại đọc tiếng Anh nên khi gặp thông báo lỗi nhiều bạn SV thường không đọc mà chỉ “sửa mò” nên mất rất nhiều thời gian. Tuy nhiên, trên thực tế số câu thông báo lỗi thường gặp không nhiều nên các bạn cần chịu khó đọc hiểu và tra từ nếu cần. Sau một hai lần thì các bạn sẽ nhớ và hiểu nên sẽ sửa lỗi rất nhanh.

Chắc chắn rằng nếu bạn không thể đọc hiểu các câu thông báo lỗi thì không thể lập trình được.

Các bạn cũng có thể tra cứu nhanh các câu thông báo lỗi được trình bày ở trang sau. Tuy nhiên các bạn vẫn nên ghi nhớ ý nghĩa của các từ tiếng Anh trong các câu thông báo lỗi để sau này có thể đọc được nhiều câu thông báo lỗi khác nữa.

STT	Câu thông báo lỗi	Ý nghĩa
1	Cannot convert parameter 1 from 'float' to 'int'	<i>Không thể chuyển đổi tham số thứ nhất từ kiểu 'float' sang kiểu 'int'</i>
2	Cannot open include file : 'Mylib.h': No such file or directory	<i>Không thể mở file bao gồm: 'Mylib.h': Không có file hay thư mục nào như thế</i>
3	End of file found before the left brace '{'	<i>Kết thúc file nằm trước dấu ngoặc móc trái '{' (do thiếu ngoặc '}' kết thúc hàm main())</i>
4	Entry point must be defined	<i>Điểm vào của chương trình phải được định nghĩa (chưa có hàm main())</i>
5	Expected constant expression	<i>Chờ đợi biểu thức hằng</i>
6	Function does not take 1 arguments	<i>Hàm không phải có 1 đối số</i>
7	Function already has a body	<i>Hàm ... đã có rồi</i>
8	'x' followed by 'void' is illegal (did you forget a ';')?	<i>'x' được theo sau bởi void là không hợp lệ (có phải bạn quên dấu ';' không?)</i>
9	'x' is not a member of struct	<i>'x' không phải là thành viên của struct</i>
10	Identifier not found	<i>Định danh không tìm thấy</i>
11	Illegal else without matching if	<i>else không ứng với if thì không hợp lệ</i>
12	main already defined in ...	<i>Hàm main() đã được định nghĩa trong ...</i>
13	Missing ': ' before 'if'	<i>Thiếu dấu ':' trước 'if'</i>
14	Newline in constant	<i>Dòng mới trong chuỗi hằng (do thiếu dấu " để đóng chuỗi)</i>
15	'Tong': must be return a value	<i>Hàm 'Tong' phải trả về một giá trị</i>
16	Left of '.HoTen' must have class/struct/union	<i>Bên trái của '.HoTen' phải là class/struct/union</i>
17	Local function definitions are illegal	<i>Định nghĩa hàm cục bộ (nằm trong hàm khác) là không hợp lệ</i>
18	'int i' : redefinition	<i>Định nghĩa lại biến i</i>
19	Undeclared identifier	<i>Định danh chưa khai báo</i>
20	'void' function returning a value	<i>Hàm 'void' trả về giá trị (là không hợp lệ)</i>

4. Một số minh họa về cách sửa lỗi cú pháp

Các ví dụ trong phần này được trình bày theo thứ tự các câu thông báo lỗi ở mục 3

Ví dụ 1:

```

1 #include <stdio.h>
2 #include <conio.h>
3 void Swap(float &a, float &b)
4 {
5     float temp = a; a = b; b = temp;
6 }
7 int main(void)
8 {
9     int a, b;
10    printf("a=");
11    scanf("%d", &a);
12    printf("b=");
13    scanf("%d", &b);
14    Swap(a,b);
15    printf("\nSau khi gọi Swap():\n");

```

Error List

Description	File	Line
error C2664: 'Swap': cannot convert parameter 1 from 'int' to 'float &'	vd3.cpp	14

Lỗi sai ở dòng 14: Hàm ‘Swap’ không thể chuyển đổi tham số thứ 1 từ kiểu ‘int’ sang kiểu ‘&float’

Cách sửa lỗi:

Cách 1: Phải sửa lại phần khai báo biến a,b ở dòng 9 có kiểu là ‘float’ để phù hợp với kiểu của tham số được truyền cho hàm Swap như khai báo ở dòng 3.

Cách 2: Phải sửa lại phần khai báo tham số được truyền cho hàm Swap ở dòng 3 là ‘&int’ . Khi đó biến temp cũng phải được khai báo kiểu int.

Ví dụ 2:

```

(Global Scope)
#include "mylib.h"
int main(void)
{
    float r, C, S;
    r = NhapSo();
    C = ChuVi(r);
    S = DienTich(r);
    printf("\n\n");
    printf("C = %-7.2f\n", C);
    printf("S = %-7.2f\n", S);
    getch();
    return 0;
}

```

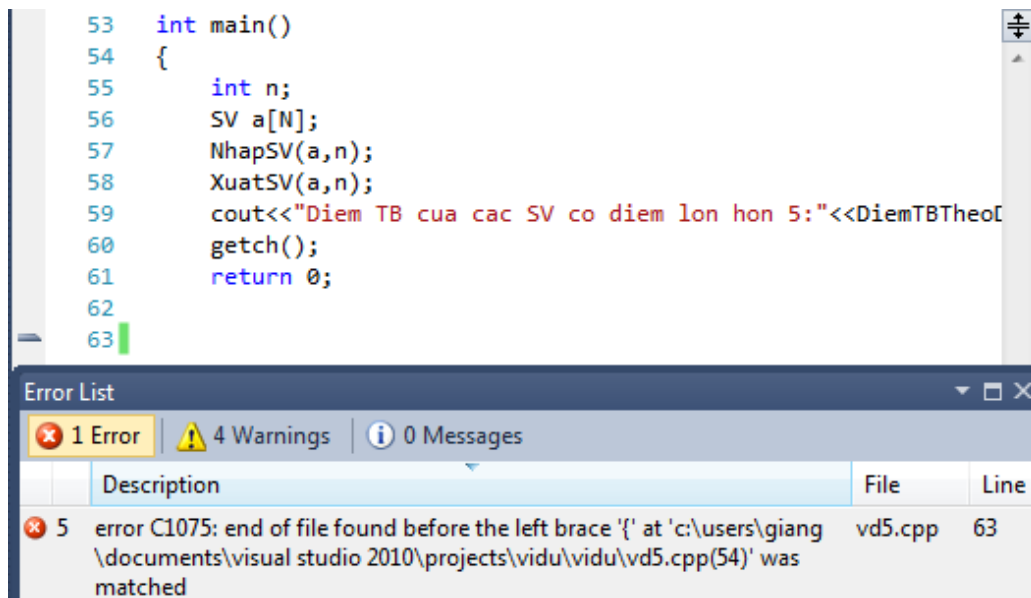
Error List

Description	File	Line
error C1083: Cannot open include file: 'Mylib.h': No such file or directory	vd2.cpp	1

Lỗi sai : Không thể mở file bao gồm: 'Mylib.h': Không có file hay thư mục nào như thế.

Cách sửa lỗi: Cần kiểm tra lại tên file 'Mylib.h' có đúng không. Nếu file này không nằm cùng thư mục với file chương trình thì phải thêm đường dẫn.

Ví dụ 3:



```
53  int main()
54  {
55      int n;
56      SV a[N];
57      NhapSV(a,n);
58      XuatSV(a,n);
59      cout<<"Diem TB cua cac SV co diem lon hon 5:"<<DiemTBTheoL
60      getch();
61      return 0;
62
63
```

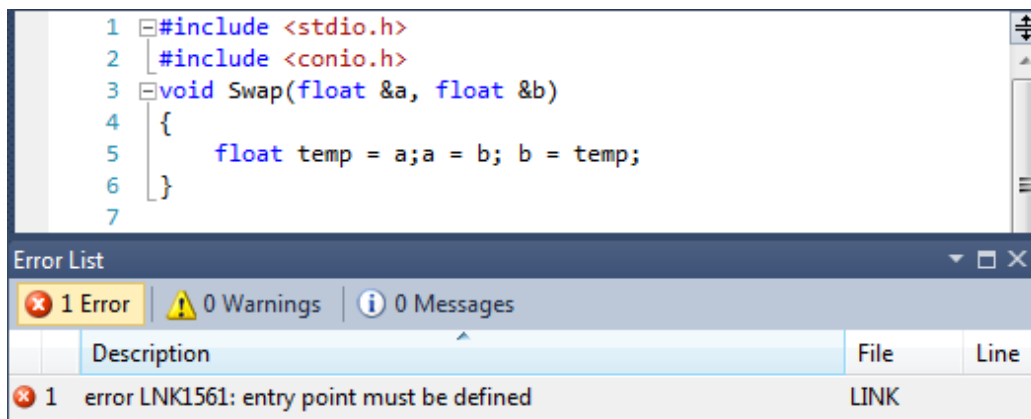
Error List

	Description	File	Line
5	error C1075: end of file found before the left brace '{' at 'c:\users\giang\documents\visual studio 2010\projects\vidu\vidu\vd5.cpp(54)' was matched	vd5.cpp	63

Lỗi sai : Kết thúc file nằm trước dấu ngoặc móc trái '{'

Cách sửa lỗi: Thêm ngoặc '}' kết thúc hàm main()

Ví dụ 4:



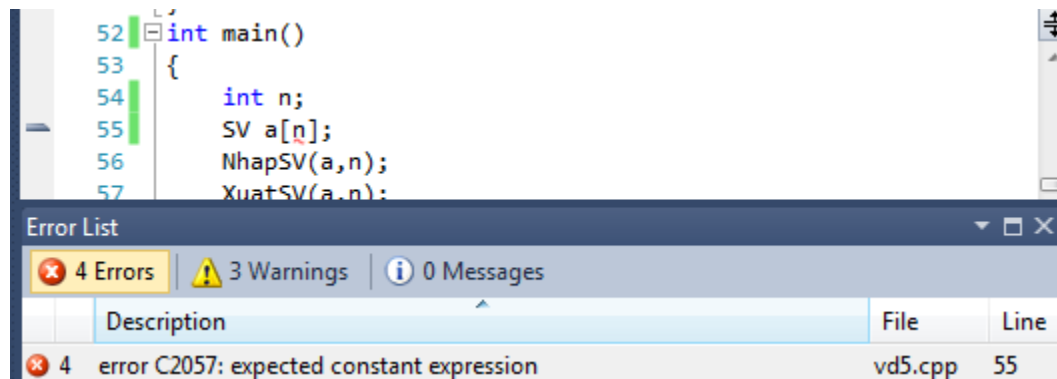
```
1  #include <stdio.h>
2  #include <conio.h>
3  void Swap(float &a, float &b)
4  {
5      float temp = a;a = b; b = temp;
6  }
7
```

Error List

	Description	File	Line
1	error LNK1561: entry point must be defined	LINK	

Lỗi sai : Điểm vào của chương trình phải được định nghĩa (chưa có hàm main())

Cách sửa lỗi: Cần bổ sung thêm hàm main()

Ví dụ 5:

```
52 int main()
53 {
54     int n;
55     SV a[n];
56     NhapSV(a,n);
57     XuatSV(a,n);
```

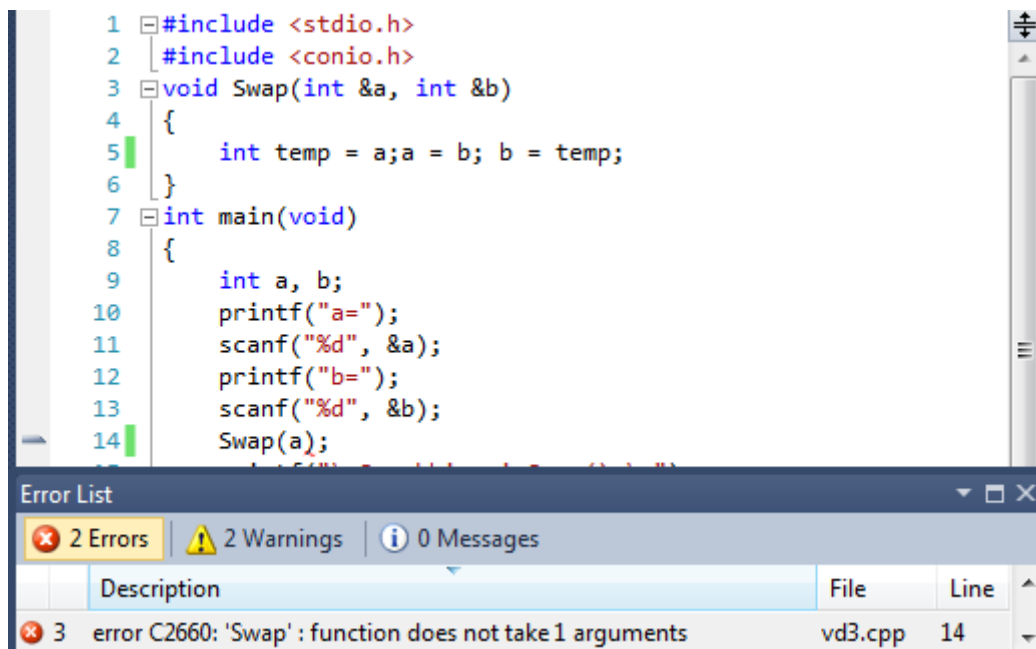
Error List

4 Errors | 3 Warnings | 0 Messages

Description	File	Line
error C2057: expected constant expression	vd5.cpp	55

Lỗi sai ở dòng 55: Chờ đợi biểu thức hằng (do ngôn ngữ C không cho phép khai báo kích thước mảng là biến)

Cách sửa lỗi: Thay kích thước mảng bằng hằng số, ví dụ : **SV a[100];**

Ví dụ 6:

```
1 #include <stdio.h>
2 #include <conio.h>
3 void Swap(int &a, int &b)
4 {
5     int temp = a; a = b; b = temp;
6 }
7 int main(void)
8 {
9     int a, b;
10     printf("a=");
11     scanf("%d", &a);
12     printf("b=");
13     scanf("%d", &b);
14     Swap(a);
```

Error List

2 Errors | 2 Warnings | 0 Messages

Description	File	Line
error C2660: 'Swap' : function does not take 1 arguments	vd3.cpp	14

Lỗi sai ở dòng 14: Hàm 'Swap' không phải có 1 tham số

Cách sửa lỗi: Bổ sung thêm tham số cho câu lệnh gọi hàm Swap() ở dòng 14:
Swap(a,b);

Ví dụ 7:

```

35 void XuatSV(SV a[],int n)
36 {
37     cout<<"STT \tMaSV \tHoTen \t\tNamSinh \tDiem \n";
38     for(int i=0;i<n;i++)
39         cout<<i+1<<"\t"<<a[i].MaSV<<"\t"<<a[i].HoTen<<"\t"<<a[i].Diem<<"\n";
40 }
41 float DiemTBTheoDK(SV a[],int n)
42 {
43     float tong=0;int dem=0;
44     for(int i=0;i<n;i++)
45         if (a[i].Diem>5)
46         {
47             tong=tong+a[i].Diem;
48             dem++;
49         }
50     return tong/dem;
51 }
52 void XuatSV(SV a[],int n)
53 {
54     cout<<"STT \tMaSV \tHoTen \t\tNamSinh \tDiem \n";
55     for(int i=0;i<n;i++)
56         cout<<i+1<<"\t"<<a[i].MaSV<<"\t"<<a[i].HoTen<<"\t"<<a[i].Diem<<"\n";
57 }

```

Error List

3 Errors 3 Warnings 0 Messages

Description	File	Line
error C2084: function 'void XuatSV(SV [],int)' already has a body	vd5.cpp	53

Lỗi sai ở dòng 53: Hàm void XuatSV() đã có rồi

Cách sửa lỗi: Xóa hàm void XuatSV() ở dòng 53

Ví dụ 8:

```

4 #define N 100
5 using namespace std;
6 typedef struct SV
7 {
8     int NamSinh;
9     float Diem;
10    char MaSV[10],HoTen[50];
11 }
12 void NhapSV(SV a[],int &n)

```

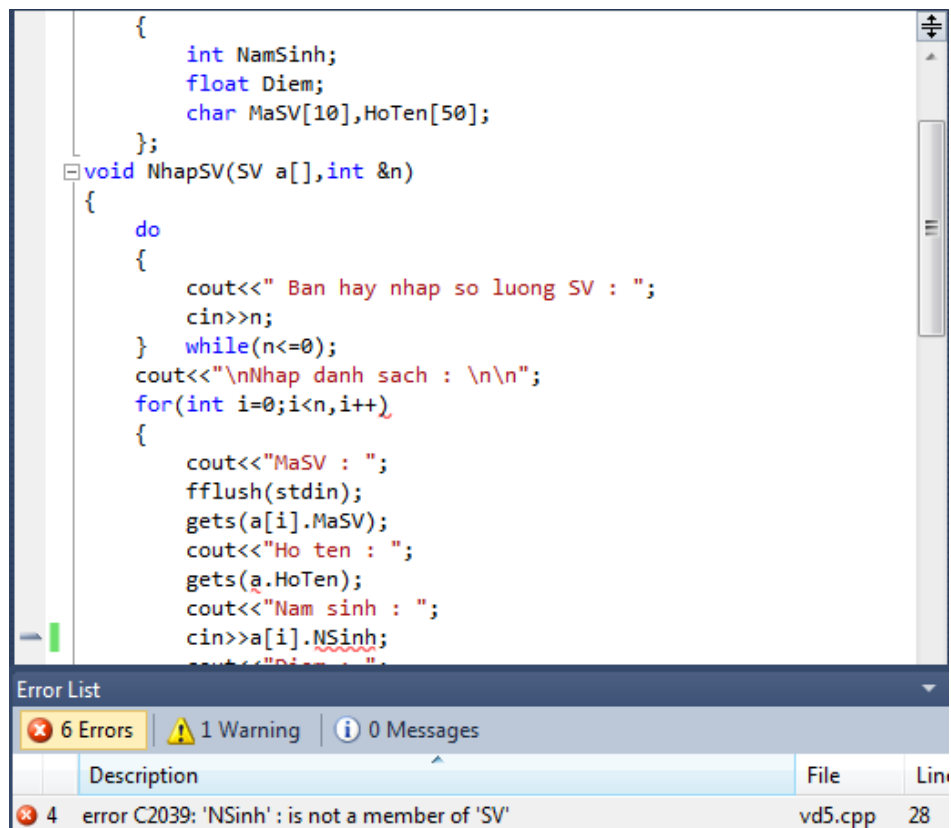
Output Error List

7 Errors 0 Warnings 0 Messages

Description	File	Line
error C2628: 'SV' followed by 'void' is illegal (did you forget a ';?')	vd5.cpp	12

Lỗi sai ở dòng 12: 'SV' được theo sau bởi 'void' thì không hợp lệ (do khai báo kiểu 'SV' chưa xong)

Cách sửa lỗi: Bổ sung thêm dấu ';' kết thúc câu lệnh khai báo kiểu SV ở cuối dòng 11.

Ví dụ 9:

```
{
    int NamSinh;
    float Diem;
    char MaSV[10], HoTen[50];
};

void NhapSV(SV a[], int &n)
{
    do
    {
        cout<<" Ban hay nhap so luong SV : ";
        cin>>n;
    } while(n<=0);
    cout<<"\nNhap danh sach : \n\n";
    for(int i=0; i<n; i++)
    {
        cout<<"MaSV : ";
        fflush(stdin);
        gets(a[i].MaSV);
        cout<<"Ho ten : ";
        gets(a[i].HoTen);
        cout<<"Nam sinh : ";
        cin>>a[i].NSinh;
    }
}
```

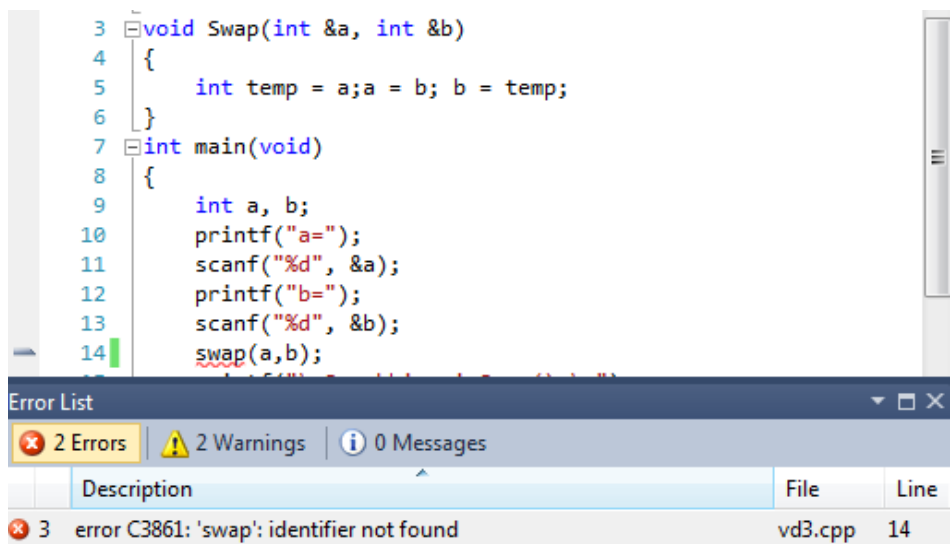
Error List

6 Errors 1 Warning 0 Messages

Description	File	Line
error C2039: 'NSinh' : is not a member of 'SV'	vd5.cpp	28

Lỗi sai ở dòng 29: 'NSinh' không phải là thành phần của struct

Cách sửa lỗi: Thay 'NSinh' bằng 'NamSinh' như phần khai báo của struct SV

Ví dụ 10:

```
3 void Swap(int &a, int &b)
4 {
5     int temp = a; a = b; b = temp;
6 }
7 int main(void)
8 {
9     int a, b;
10    printf("a=");
11    scanf("%d", &a);
12    printf("b=");
13    scanf("%d", &b);
14    swap(a,b);
}
```

Error List

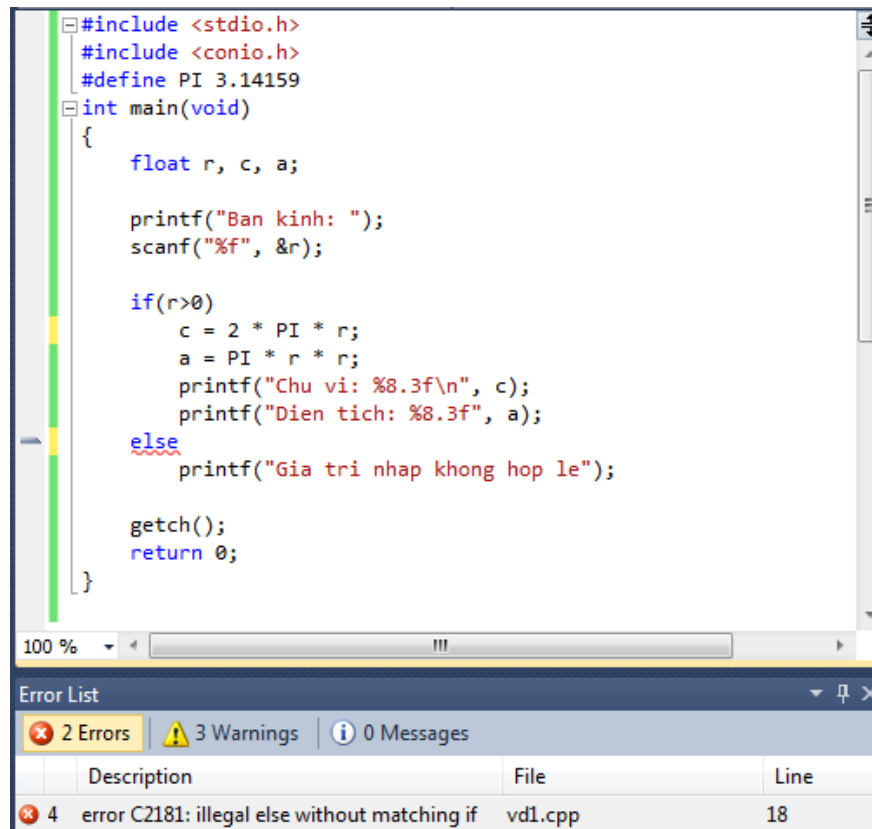
2 Errors 2 Warnings 0 Messages

Description	File	Line
error C3861: 'swap': identifier not found	vd3.cpp	14

Lỗi sai ở dòng 14: ‘SWAP’ là định danh không tìm thấy

Cách sửa lỗi: Thay ‘SWAP’ bằng ‘Swap’ giống như tên hàm đã cài đặt ở trên

Ví dụ 11:



```
#include <stdio.h>
#include <conio.h>
#define PI 3.14159
int main(void)
{
    float r, c, a;

    printf("Ban kinh: ");
    scanf("%f", &r);

    if(r>0)
        c = 2 * PI * r;
        a = PI * r * r;
        printf("Chu vi: %8.3f\n", c);
        printf("Dien tich: %8.3f", a);
    else
        printf("Gia tri nhap khong hop le");

    getch();
    return 0;
}
```

100 %

Error List

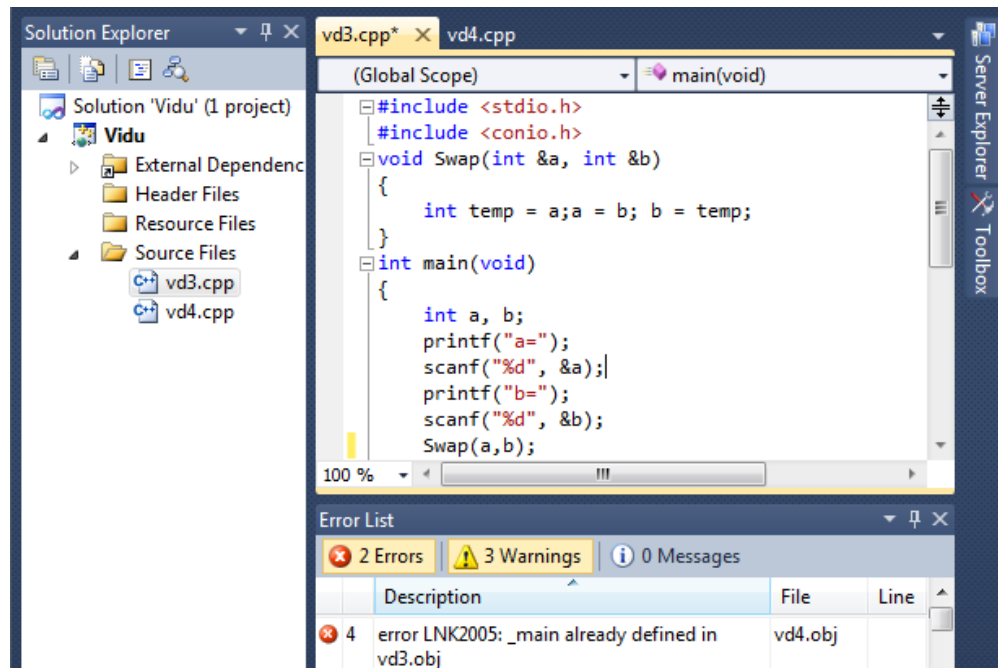
2 Errors | 3 Warnings | 0 Messages

	Description	File	Line
4	error C2181: illegal else without matching if	vd1.cpp	18

Lỗi sai ở dòng 18: else không ứng với if thì không hợp lệ

Cách sửa lỗi: Thêm dấu {} trong câu lệnh if phía trên else (vì if có 4 câu lệnh con nên cần có ngoặc)

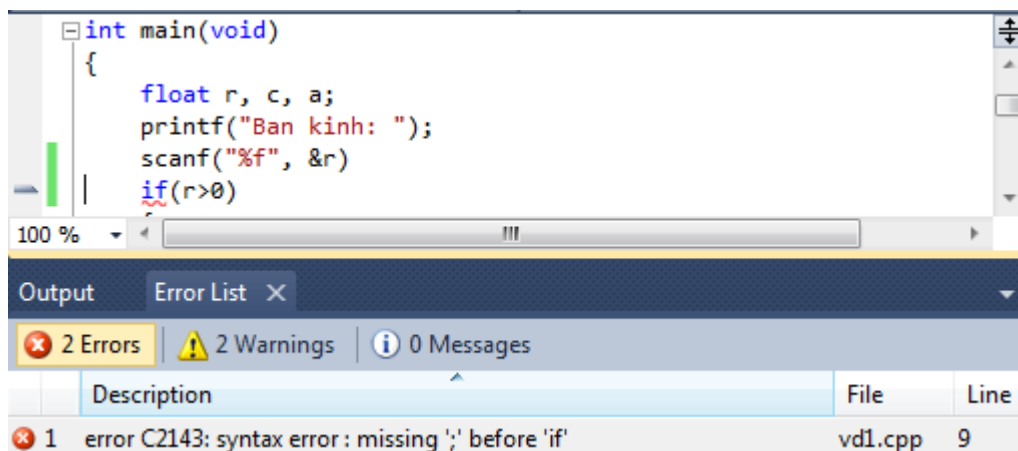
Ví dụ 12:



Lỗi sai : Hàm `main()` đã được định nghĩa

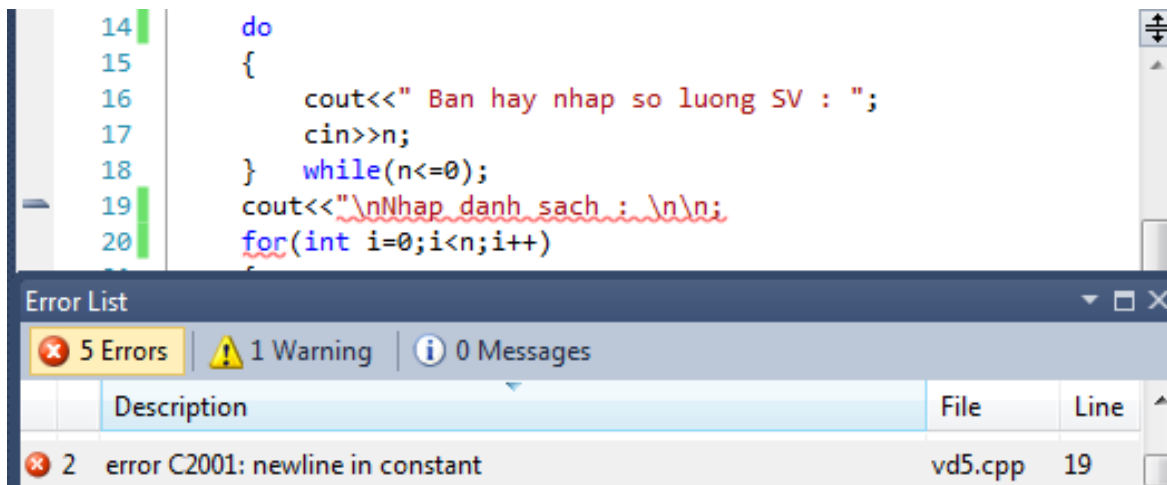
Cách sửa lỗi: Cách sửa lỗi: Dời chỗ (remove) file chương trình `vd4.cpp` (Nhấp phải lên file `vd4.cpp` và chọn remove). Lưu ý, tại mỗi thời điểm chỉ cho phép thực thi một hàm `main()` nên không được để nhiều hơn một file chương trình (.cpp) trong mỗi project khi thực thi chương trình.

Ví dụ 13:



Lỗi sai ở dòng 9: Thiếu dấu `;` trước `if`

Cách sửa lỗi: Thêm dấu `;` ở cuối dòng 9

Ví dụ 14:

```
14 do
15 {
16     cout<<" Ban hay nhap so luong SV : ";
17     cin>>n;
18 } while(n<=0);
19 cout<<"\nNhap danh sach : \n\n";
20 for(int i=0;i<n;i++)
```

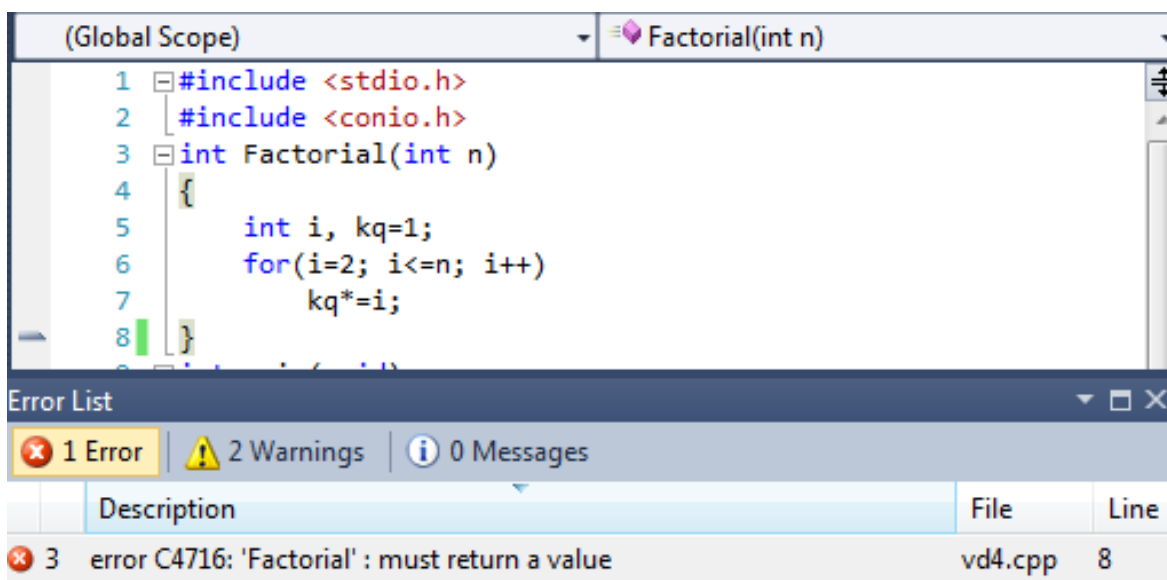
Error List

5 Errors | 1 Warning | 0 Messages

Description	File	Line
error C2001: newline in constant	vd5.cpp	19

Lỗi sai ở dòng 19: Dòng mới trong (chuỗi) hằng

Cách sửa lỗi: Thêm dấu “ ” để đóng chuỗi ở cuối dòng 19

Ví dụ 15:

```
(Global Scope) | Factorial(int n)
1 #include <stdio.h>
2 #include <conio.h>
3 int Factorial(int n)
4 {
5     int i, kq=1;
6     for(i=2; i<=n; i++)
7         kq*=i;
8 }
```

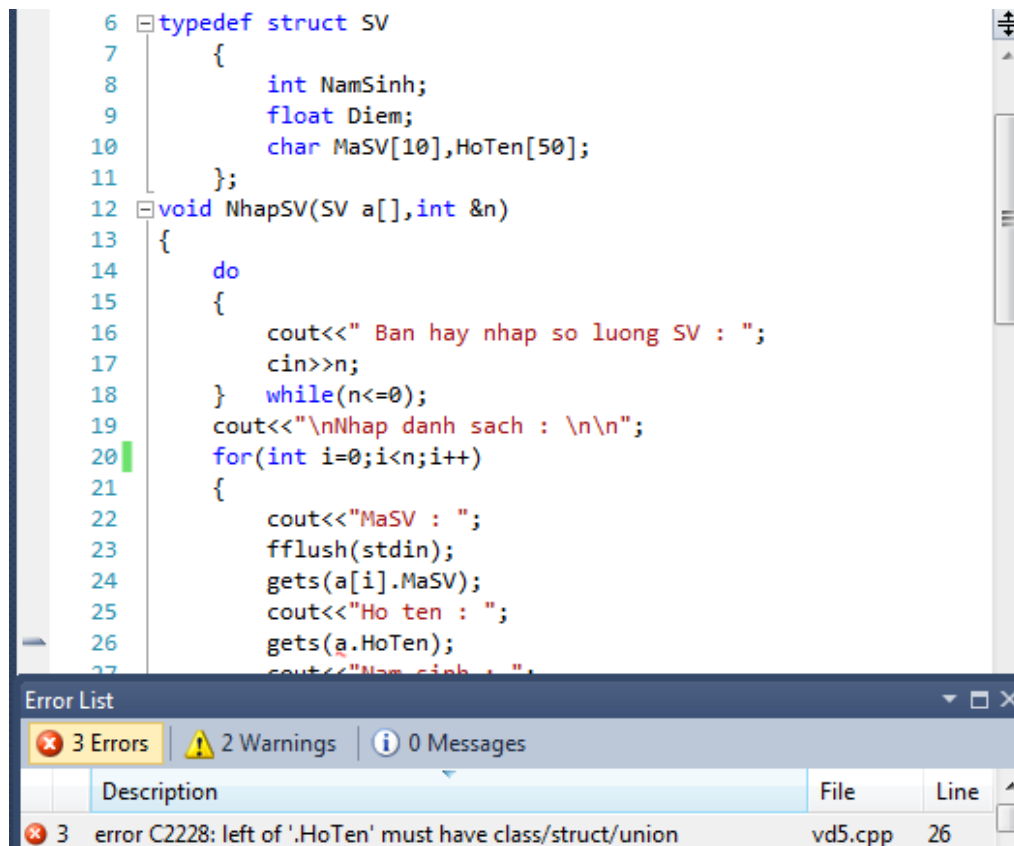
Error List

1 Error | 2 Warnings | 0 Messages

Description	File	Line
error C4716: 'Factorial' : must return a value	vd4.cpp	8

Lỗi sai ở dòng 8: Hàm ‘Factorial’ phải trả về giá trị (do dòng 3 đã khai báo hàm này trả về kiểu int)

Cách sửa lỗi: Bổ sung câu lệnh: return kq; ở cuối hàm.

Ví dụ 16:

```
6 typedef struct SV
7 {
8     int NamSinh;
9     float Diem;
10    char MaSV[10],HoTen[50];
11 };
12 void NhapSV(SV a[],int &n)
13 {
14     do
15     {
16         cout<<" Ban hay nhap so luong SV : ";
17         cin>>n;
18     } while(n<=0);
19     cout<<"\nNhap danh sach : \n\n";
20     for(int i=0;i<n;i++)
21     {
22         cout<<"MaSV : ";
23         fflush(stdin);
24         gets(a[i].MaSV);
25         cout<<"Ho ten : ";
26         gets(a.HoTen);
27         cout<<"Nam sinh : ";
```

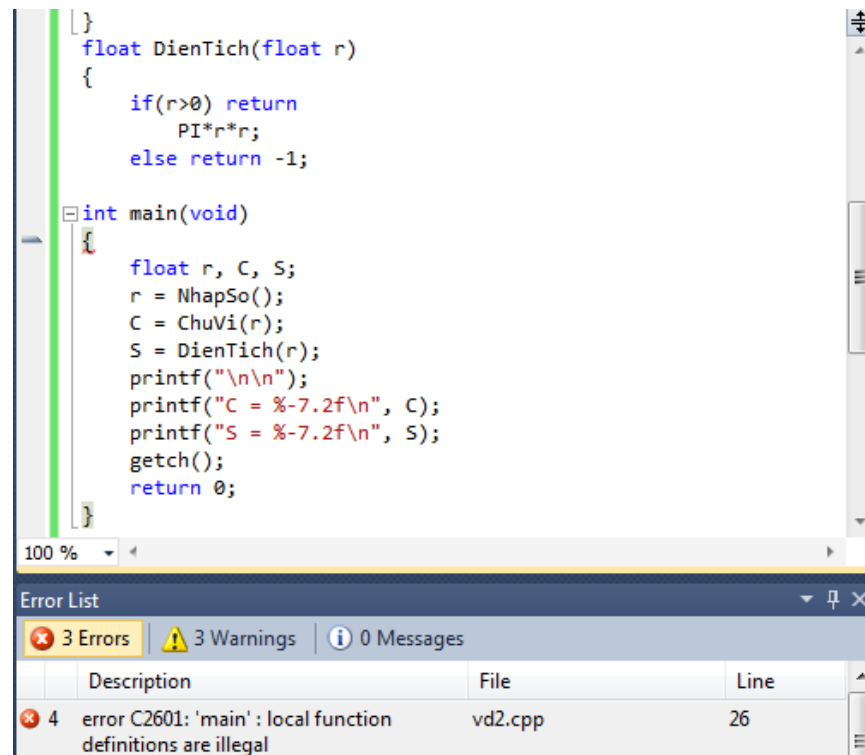
Error List

3 Errors 2 Warnings 0 Messages

Description	File	Line
error C2228: left of '.HoTen' must have class/struct/union	vd5.cpp	26

Lỗi sai ở dòng 26: Bên trái của '.Hoten' phải là class/struct/union (trong trường hợp này là struct vì HoTen là 1 thành phần của struct SV)

Cách sửa lỗi: Sửa lại là: gets(a[i].Hoten); (a là biến lưu địa chỉ của mảng a chứa danh sách sinh viên chứ không phải là biến lưu thông tin của một sinh viên)

Ví dụ 17:

```

}
float DienTich(float r)
{
    if(r>0) return
        PI*r*r;
    else return -1;

    int main(void)
    {
        float r, C, S;
        r = NhapSo();
        C = ChuVi(r);
        S = DienTich(r);
        printf("\n\n");
        printf("C = %-7.2f\n", C);
        printf("S = %-7.2f\n", S);
        getch();
        return 0;
    }
}

```

100 %

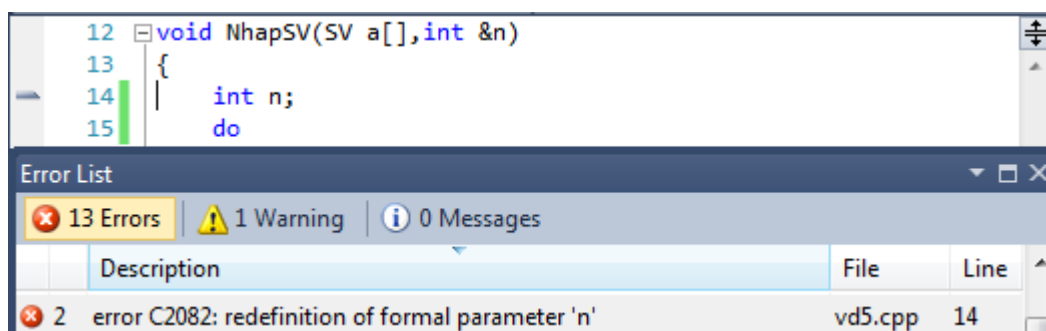
Error List

3 Errors 3 Warnings 0 Messages

	Description	File	Line
4	error C2601: 'main': local function definitions are illegal	vd2.cpp	26

Lỗi sai ở dòng 26: ‘main’ : định nghĩa hàm cục bộ (nằm trong hàm khác) là không hợp lệ.

Cách sửa lỗi: Do hàm main() đang nằm trong hàm DienTich() nên cần đóng ngoặc kết thúc hàm DienTich()

Ví dụ 18:

```

12 void NhapSV(SV a[],int &n)
13 {
14     int n;
15     do

```

Error List

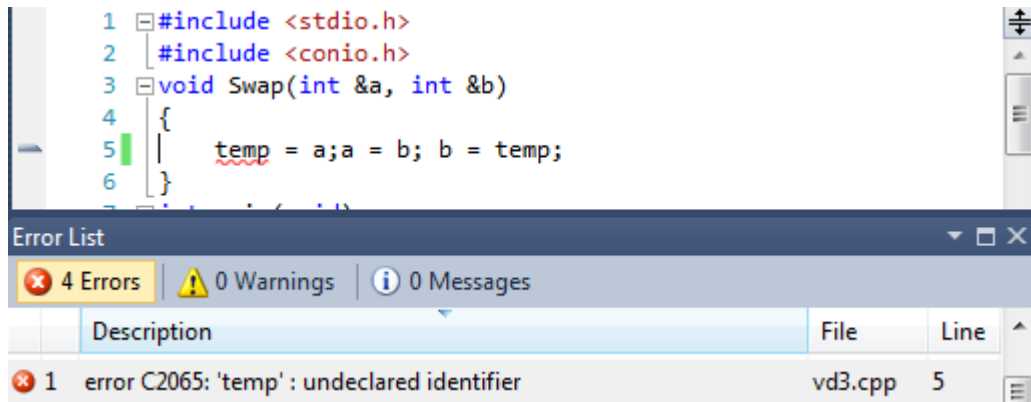
13 Errors 1 Warning 0 Messages

	Description	File	Line
2	error C2082: redefinition of formal parameter 'n'	vd5.cpp	14

Lỗi sai ở dòng 14: Định nghĩa lại tham số hình thức ‘n’

Cách sửa lỗi: Bỏ khai báo lại biến n ở dòng 14.

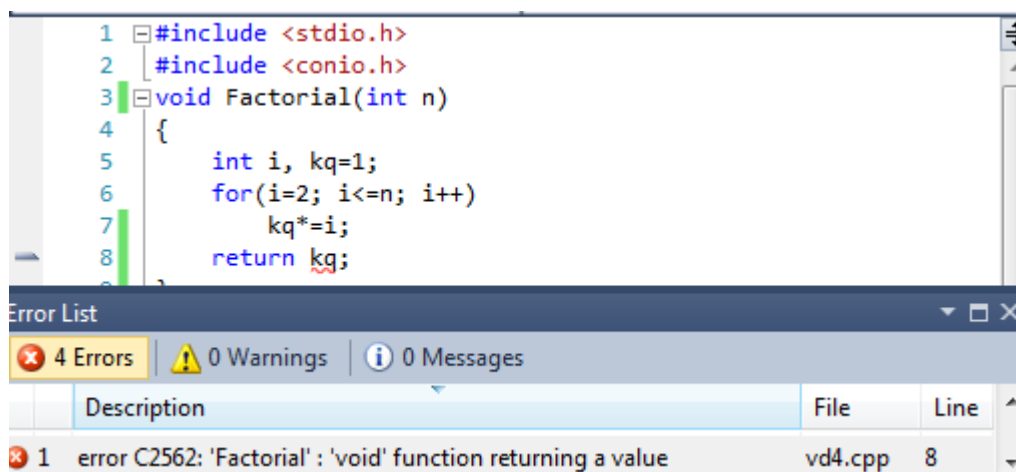
Ví dụ 19:



Lỗi sai ở dòng 5: ‘temp’ chưa khai báo

Cách sửa lỗi: Khai báo biến ‘temp’

Ví dụ 20:



Lỗi sai ở dòng 8: Hàm ‘Factorial’ là hàm void mà trả về giá trị

Cách sửa lỗi: Phải sửa lại kiểu trả về của hàm là int (trong phần khai báo ở dòng 3)

II. CÁCH SỬA LỖI THỜI GIAN CHẠY

Các lỗi thời gian chạy có thể xảy ra vì các lý do sau đây:

- + Khi chương trình cố gắng thực hiện một hoạt động bất hợp pháp như phân chia một số cho số không.
- + Việc nhập dữ liệu không thành công.

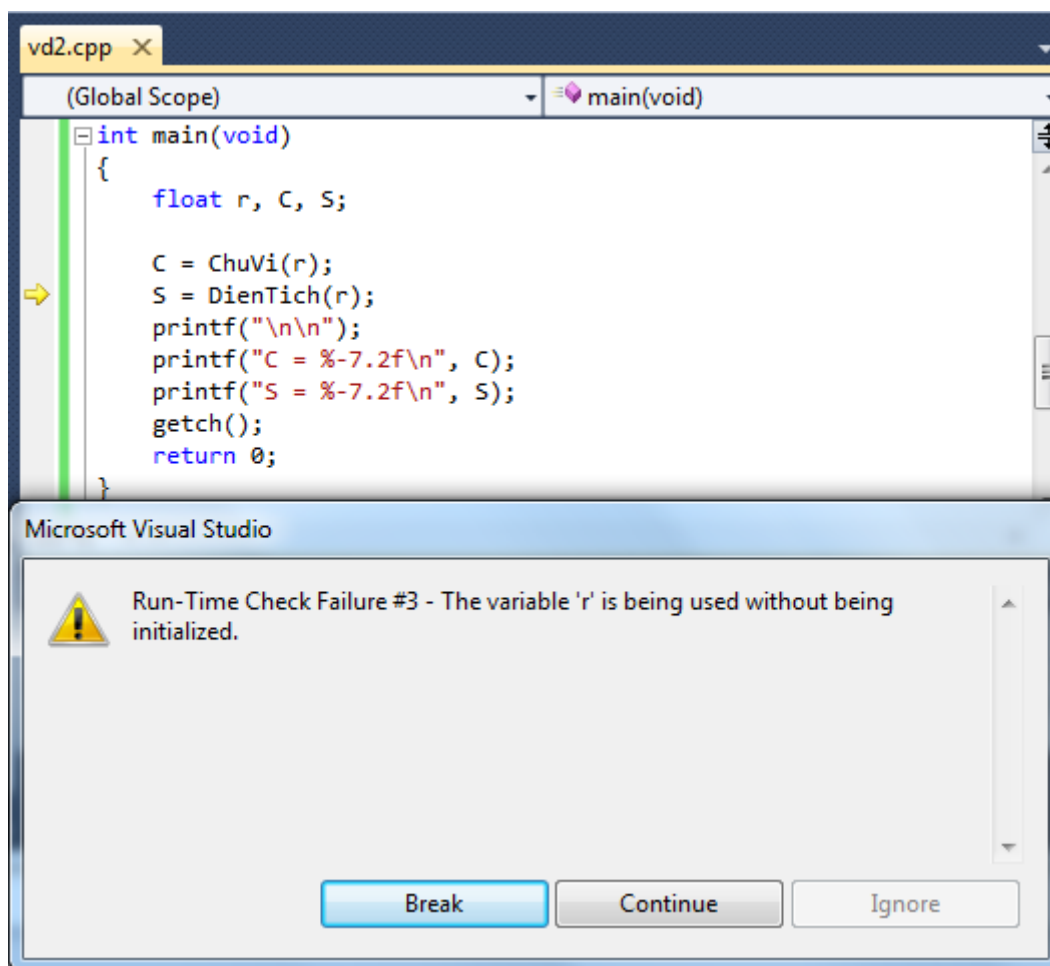
+ Xảy ra vấn đề phân cứng như lỗi đĩa cứng hoặc ổ đĩa cứng hoặc lỗi máy in,...

1. Cách sửa lỗi thời gian chạy

- Nhấp chọn nút Break, chương trình dịch sẽ chỉ ra dòng bị lỗi tương ứng trong chương trình. Ta cần đọc hiểu câu thông báo lỗi để sửa lỗi nhanh.

2. Các ví dụ minh họa về lỗi thời gian chạy

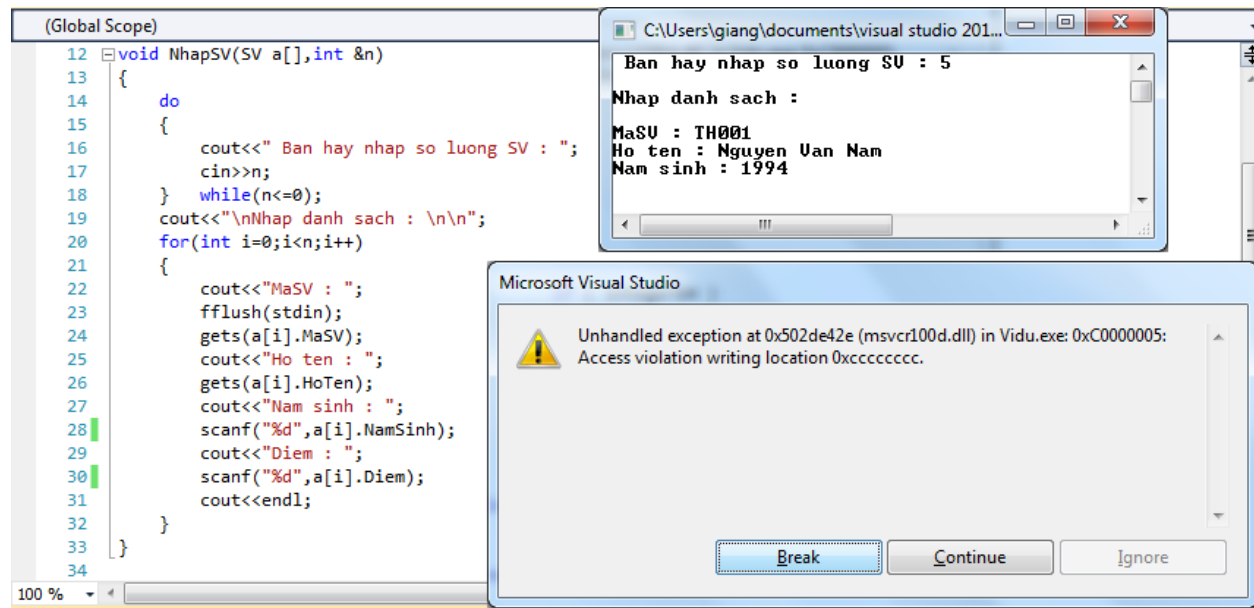
Ví dụ 1:



Lỗi sai: Biến 'r' đang được sử dụng mà chưa khởi tạo (giá trị ban đầu)

Cách sửa lỗi: Cần bổ sung câu lệnh nhập giá trị cho biến r trước khi gọi hàm ChuVi() và DienTich()

Ví dụ 2:



Lỗi sai: Lỗi ngoài tầm kiểm soát, ghi vào địa chỉ không được phép.

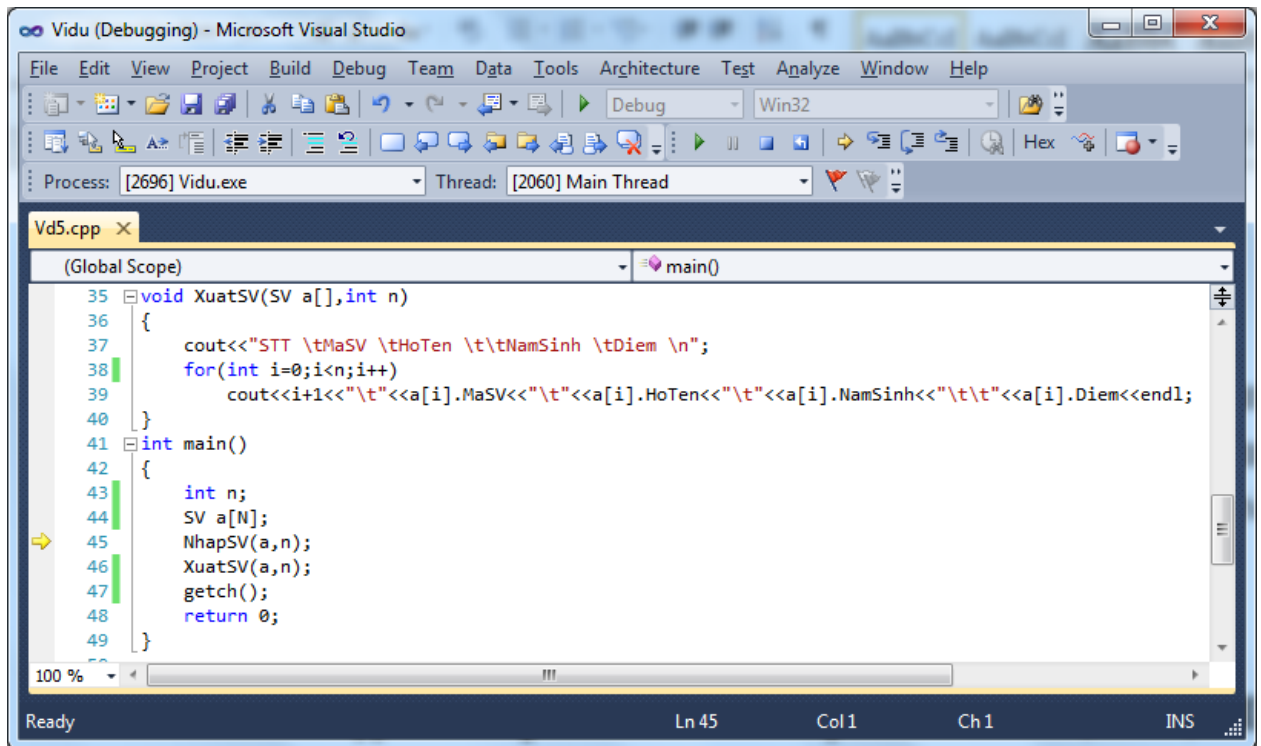
Cách sửa lỗi: Khi chương trình chạy xong câu lệnh nhập năm sinh của SV ở dòng 28 thì bị lỗi nên dễ thấy câu lệnh này bị sai cú pháp của hàm scanf(). Cần bổ sung toán tử & trước biến cần nhập: scanf("%d", &a[i].NamSinh);

III. CÁCH SỬA LỖI LOGIC

Phần này sẽ trình bày cách sửa lỗi logic bằng công cụ Debug.

1. Chạy từng lệnh

Khi bắt đầu chạy debug từng lệnh (bằng cách nhấn phím F10 hay F11) thì điều khiển của chương trình (hiển thị bằng mũi tên màu vàng ở cột mốc trái của vùng soạn thảo) sẽ chạy từng dòng lệnh và luôn luôn bắt đầu từ hàm main() như hình sau:



a. Chạy lướt qua hàm: nhấn phím F10 hoặc chọn Debug → Step Over

Nếu dòng hiện tại có lời gọi đến một hàm nào đó trong chương trình thì điều khiển của chương trình sẽ lướt qua hàm này chứ không đi vào để chạy từng lệnh trong hàm này.

b. Đi vào hàm: nhấn phím F11 hoặc chọn Debug → Step Into

Nếu dòng hiện tại có lời gọi đến một hàm nào đó thì điều khiển sẽ đi vào để chạy từng lệnh trong hàm này.

Hiển nhiên, nếu dòng hiện tại không chứa lời gọi đến một hàm nào khác thì nhấn phím F10 hay F11 là như nhau.

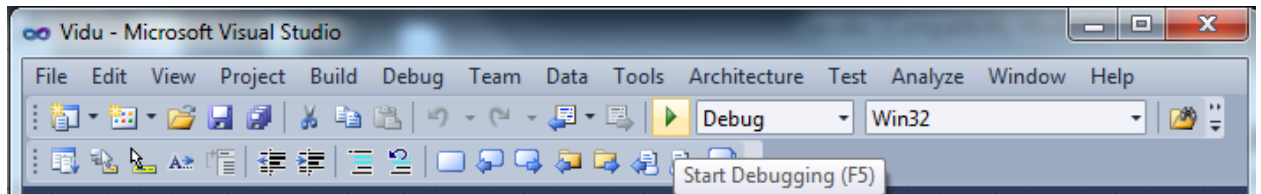
c. Chạy nhanh ra khỏi hàm (mà điều khiển của chương trình đang dừng): nhấn tổ hợp phím **Shift+F11** hoặc chọn **Debug → Step Out**.

2. Dừng chương trình tại các điểm dừng

Nếu bạn không muốn mất thời gian chạy từng dòng lệnh của toàn bộ chương trình mà chỉ cần kiểm tra chương trình ở một vài lệnh hay thời điểm cần thiết nào

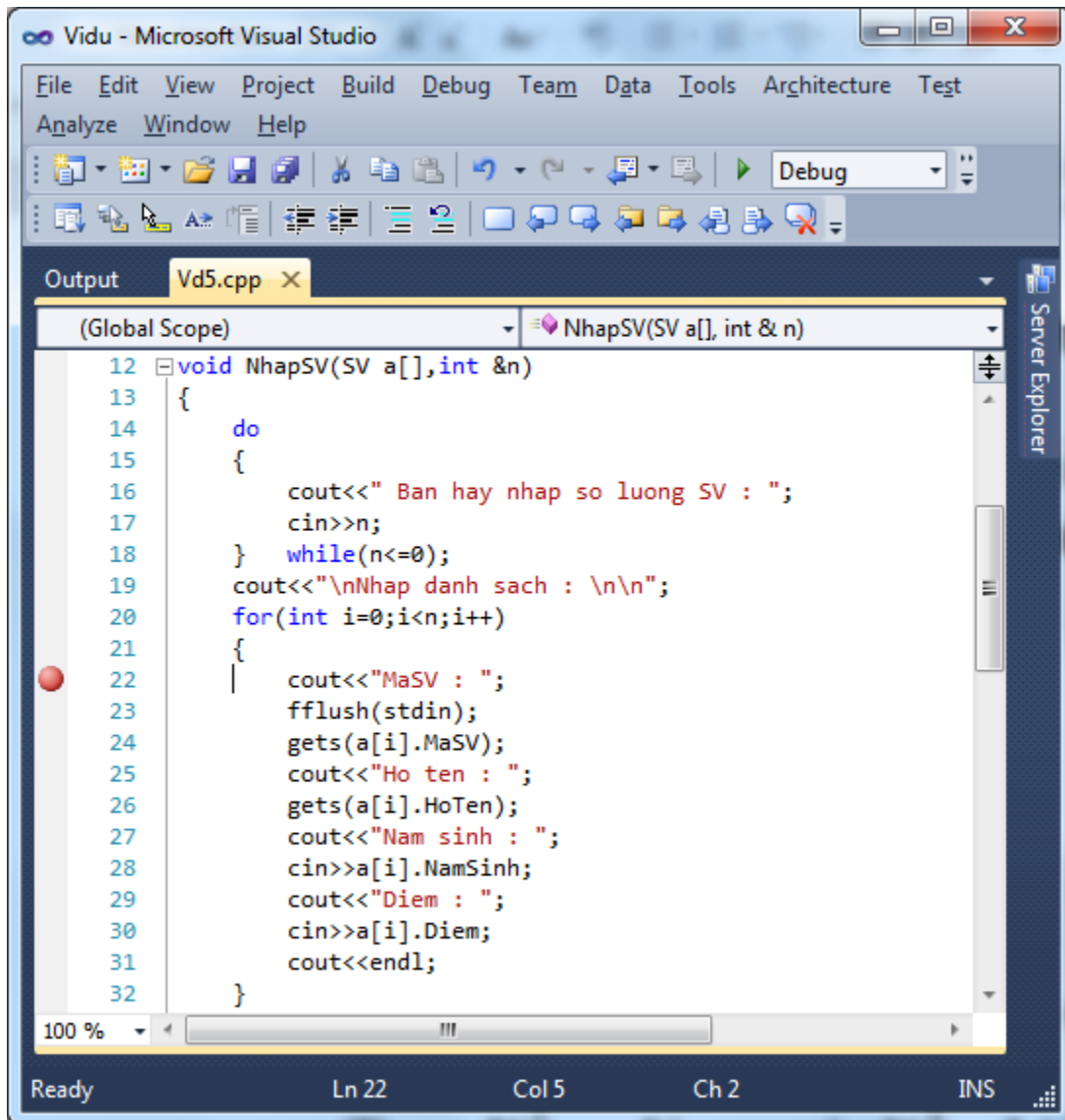
đó thì ta sẽ tạo các bạn có thể tạo các điểm dừng (break point) để kiểm tra giá trị các biến hay cách thức chương trình chạy tại các điểm này.

Sau khi tạo điểm dừng bạn nhấn F5 hay phím Start Debugging, điều khiển của chương trình sẽ dừng tại breakpoint mà bạn đã tạo



a. Cách tạo một breakpoint:

Nhấp chuột vào cột móc trái tại hàng tương ứng, khi đó, tại hàng tương ứng trong cột móc trái sẽ có một nút tròn màu đỏ như trong hình sau:

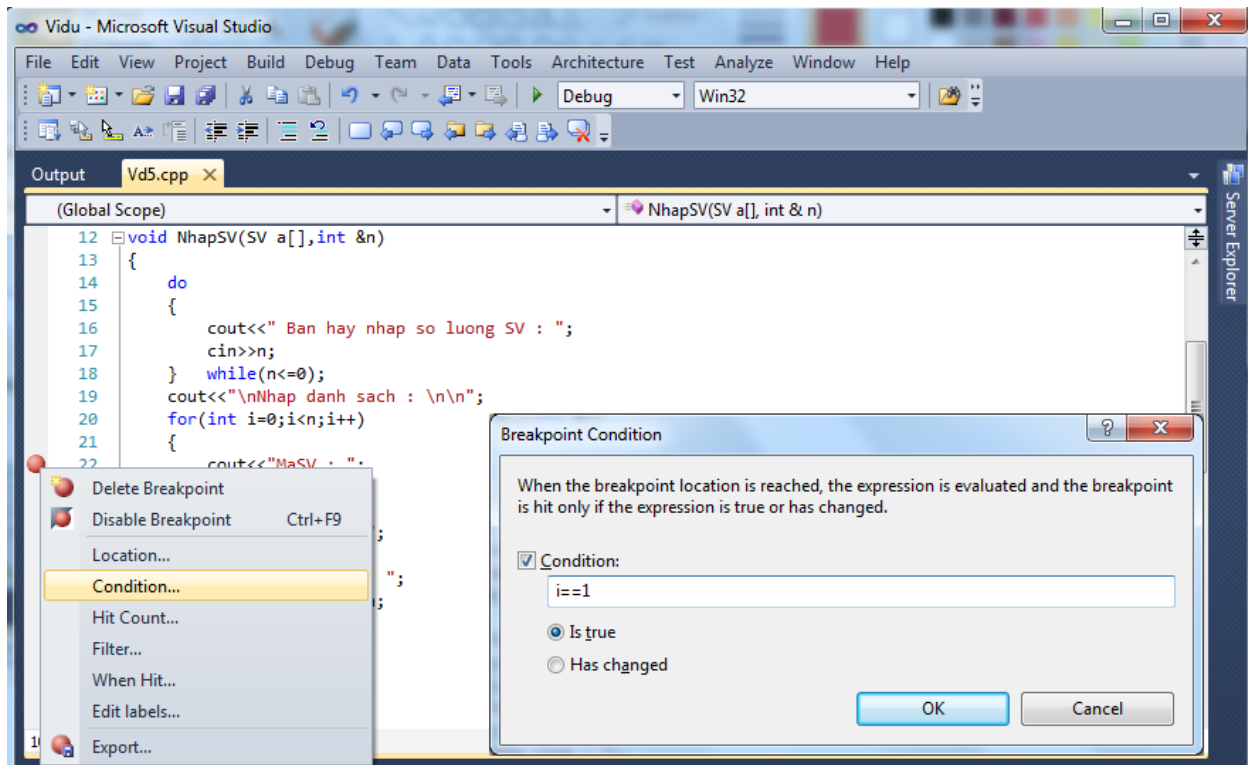


b. Hủy một breakpoint: Nhấp chuột vào breakpoint muốn hủy

c. Tạo điểm dừng có điều kiện (conditional breakpoint):

Nếu ta chỉ muốn chương trình sẽ dừng tại lần lặp hay lần gọi đệ quy cụ thể nào đó: Tạo một breakpoint tại hàng muốn dừng sau đó nhấp phải chuột phải tại breakpoint này và chọn **Condition** . Sau đó nhập điều kiện vào vùng condition.

Trong hình sau tôi tạo một breakpoint trong vòng lặp tại lần lặp thỏa điều kiện là `i==1`.



d. Dừng chương trình. Nếu không muốn chạy Debug tiếp nữa: Nhấn tổ hợp phím **Shift+F5** hoặc chọn **Debug** → **Stop Debugging** để dừng chương trình.

3. Xem giá trị các biến khi debug

Việc xem giá trị các biến tại một thời điểm nào đó khi thực thi chương trình sẽ giúp ta kiểm tra xem giá trị các biến này có đúng không và cách thức chương trình thực hiện các lệnh mà ta đã viết để tìm ra nguyên nhân làm chương trình cho kết quả sai.

Ta có thể xem giá trị các biến bằng một trong 3 cửa sổ:

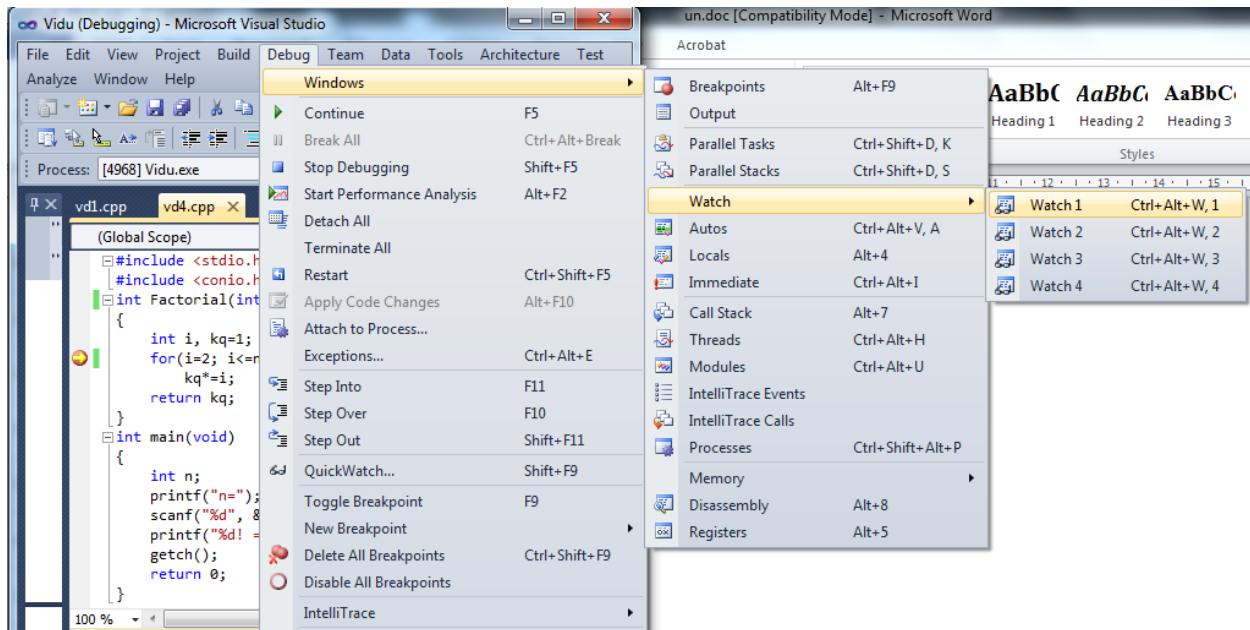
- **Autos:** Hiện thị giá trị tất cả các biến có trong dòng lệnh hiện tại và dòng lệnh trước đó.
- **Locals:** Hiện thị giá trị tất cả các biến cục bộ trong hàm hiện tại
- **Watch:** Cho phép xem giá trị các biến hay biểu thức bất kỳ mà bạn muốn

Trong trường hợp các cửa sổ này không có sẵn chúng ta có thể mở bằng cách:

a. Cách mở cửa sổ Autos: nhấn tổ hợp phím **Ctrl+Alt+V,A** hoặc chọn **Debug → Windows → Autos**

b. Cách mở cửa sổ Local: nhấn tổ hợp phím **Alt+4** hoặc chọn **Debug → Windows → Locals**

c. Cách mở cửa sổ watch: chọn **Debug → Windows → Watch → Watch1**



4. Ví dụ tìm lỗi logic bằng công cụ Debug

Ví dụ: Cho một mảng a chứa n số thực, ta cần viết hàm tìm giá trị nhỏ nhất trong các số dương của mảng.

Nếu chỉ cần tìm giá trị nhỏ nhất (Min) của một dãy số, ta có 2 bước:

Bước 1: Gán giá trị Min ban đầu (thường gán là giá trị phần tử đầu tiên của mảng tức là $a[0]$)

Bước 2: So sánh giá trị Min với các giá trị còn lại của mảng ($a[i]$, $i=1, \dots, n-1$). Nếu thấy giá trị $a[i]$ nào nhỏ hơn Min thì đổi lại giá trị Min bằng $a[i]$ đó.

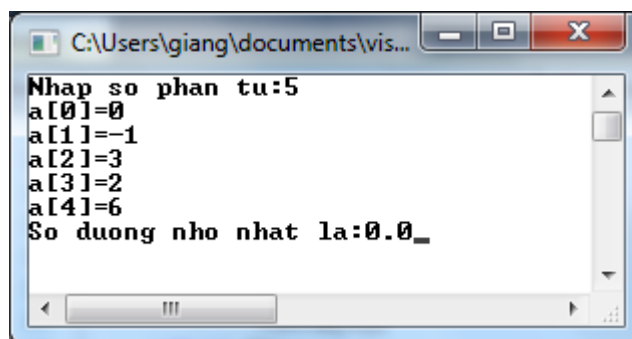
Hàm cài đặt như sau:

```
float MinMang(float a[],int n)
{
    float Min=a[0];
    for(int i=0;i<n;i++)
        if(a[i]<Min)
            Min=a[i];
    return Min;
}
```

Vậy để tìm Min của các số dương trong mảng, **thông thường các SV sẽ viết hàm như sau:**

```
float MinSoDuong(float a[],int n)
{
    float Min=a[0];
    for(int i=0;i<n;i++)
        if(a[i]>0&& a[i]<Min)
            Min=a[i];
    return Min;
}
```

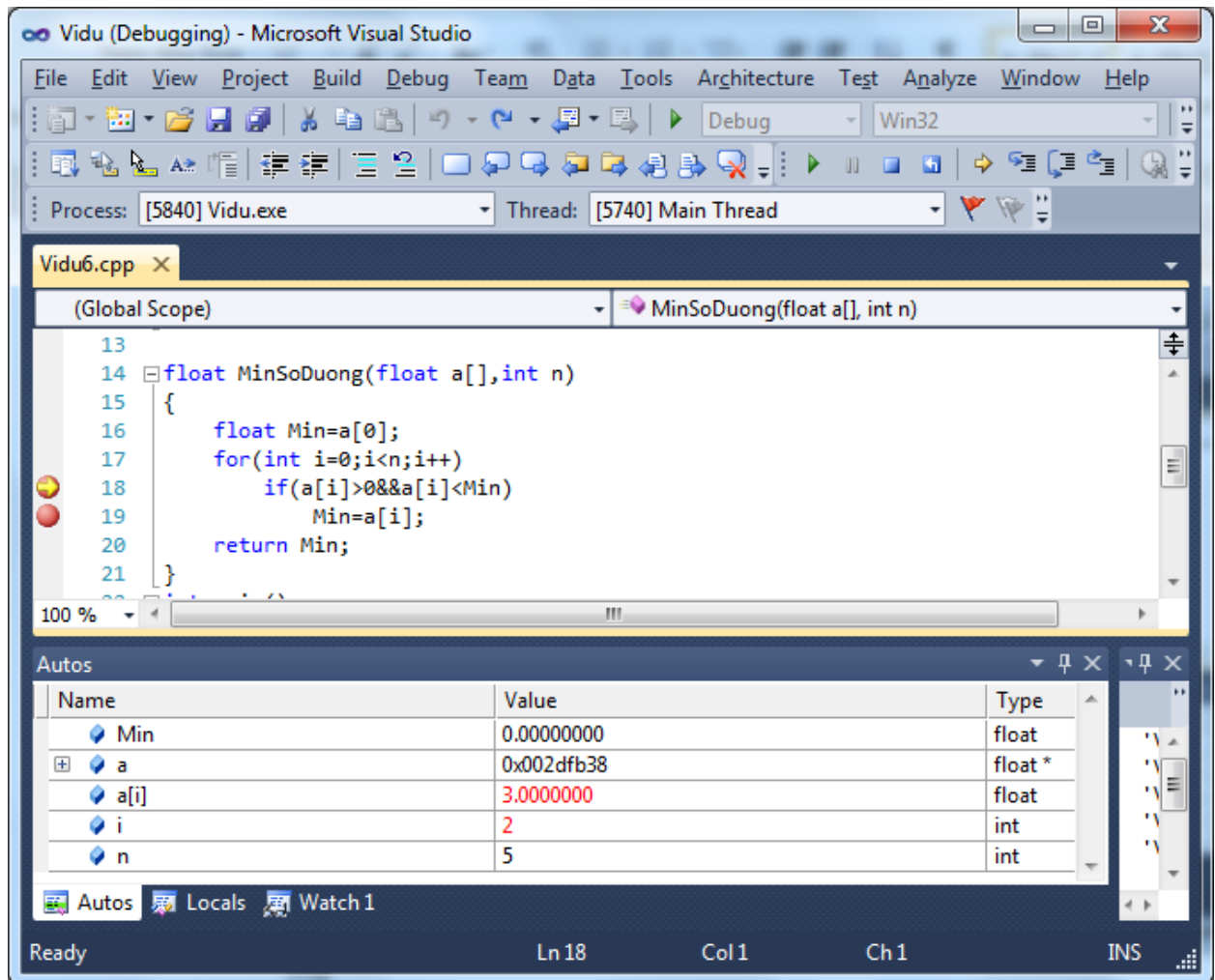
Tuy nhiên khi chạy thử kết quả sẽ bị sai (kết quả đúng là số 2):



Dĩ nhiên chương trình dịch sẽ không thông báo lỗi này vì là lỗi sai về logic của giải thuật. Vì vậy để dò tìm chúng ta có thể tự kiểm tra lại hàm MinSoDuong () bằng cách tự nhẩm hay viết ra nháp kết chạy từng câu lệnh trên dãy số cho trước. Tuy nhiên nếu thấy khó khăn, ta có thể dùng công cụ Debug để tìm nhanh ra nguyên nhân mà chương trình cho kết quả sai.

Quá trình Debug gồm các bước:

Bước 1: Xác định câu lệnh cần kiểm tra. Trong ví dụ này là câu lệnh ở dòng 18, 19 vì đây là câu lệnh mấu chốt ảnh hưởng đến giá trị của biến Min. sau đó chạy từng lệnh (F10/F11) đến lệnh cần kiểm tra hoặc chọn luôn điểm dừng (breakpoint) tại dòng 18 và 19.



Bước 2: Nhấn F5 hay nhấp chọn Start Debugging để thực thi chương trình. Điều khiển chương trình sẽ dừng ở dòng lệnh có Breakpoint là dòng 18.

Bước 3: Chọn cửa sổ cần xem giá trị các biến. Kiểm tra sự thay đổi giá trị các biến. Trong ví dụ này tôi chọn cửa sổ Autos và quan sát giá trị biến Min thay đổi theo $a[i]$ như thế nào.

Bước 4: Nhấn F5 hay nhấp chọn Start Debugging nhiều lần, mỗi lần như thế điều khiển của chương trình sẽ nhảy đến Breakpoint kế tiếp.

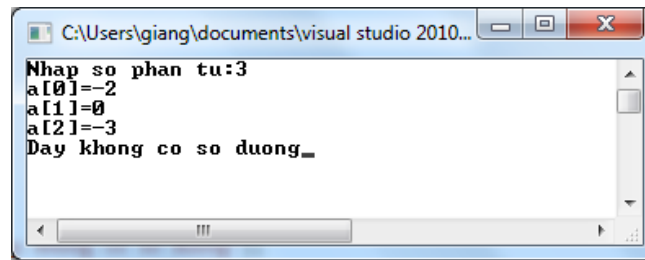
Trong trường hợp này ta nhận thấy ở tất cả các lần lặp, điều khiển của chương trình không bao giờ nhảy xuống dòng 19 tức là chương trình không thực hiện câu lệnh gán: $a[i]=Min$. Vì thế Min luôn bằng 0. Lý do duy nhất là điều kiện $a[i]<Min$ không được thỏa (vì Min ban đầu được gán là 0 nên luôn nhỏ hơn các số $a[i]$ dương).

Từ sự phân tích ở trên ta kết luận việc gán $Min=a[0]$ ($=0$) như dòng 16 là sai. Ta cần phải gán Min là giá trị dương đầu tiên của dãy (trong ví dụ này là số 3).

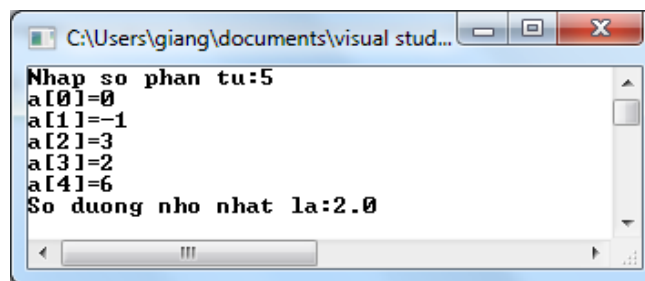
Vì thế chương trình được sửa lại như sau:

```
#include<stdio.h>
#include<conio.h>
void NhapMang(float a[],int &n)
{
    printf("Nhap so phan tu:");
    scanf("%d",&n);
    for (int i=0;i<n;i++)
    {
        printf("a[%d]=",i);
        scanf("%f",&a[i]);
    }
}
float MinSoDuong(float a[],int n)
{
    float Min=-1; int k;
    //Gán Min là số dương đầu tiên
    for(k=0;k<n;k++)
        if(a[k]>0)
        {
            Min=a[k];
            break;
        }
    //So sánh Min với các số dương còn lại và gán Min là
    //số nhỏ hơn
    for(int i=k+1;i<n;i++)
        if(a[i]>0&&a[i]<Min)
            Min=a[i];

    return Min;
    //Nếu dãy không có số dương hàm sẽ trả về giá trị Min
    //gán ban đầu là -1
}
int main()
{
    float a[100],t;
    int n;
    NhapMang(a,n);
    t=MinSoDuong(a,n);
    if (t==-1) printf("Day khong co so duong");
    else printf("So duong nho nhat la:%.1f",t);
    getch();
    return 0;
}
```



```
C:\Users\giang\documents\visual studio 2010...
Nhap so phan tu:3
a[0]=-2
a[1]=0
a[2]=-3
Day khong co so duong_
```



```
C:\Users\giang\documents\visual stud...
Nhap so phan tu:5
a[0]=0
a[1]=-1
a[2]=3
a[3]=2
a[4]=6
So duong nho nhat la:2.0
```

PHẦN 3

MỘT SỐ TỪ/CỤM TỪ TIẾNG ANH THƯỜNG GẶP TRONG CÁC CÂU THÔNG BÁO LỖI

STT	Từ/ cụm từ	Ý nghĩa	Ví dụ
1	access	truy cập	Access denied: Truy cập bị từ chối
2	already	đã ...rồi	main already defined: Hàm <i>main</i> đã định nghĩa rồi
3	argument	đối số	Function does not take 1 arguments: Hàm không phải có 1 đối số
4	constant	hằng số	Constant expression: Biểu thức hằng
5	convert	biến đổi, chuyển đổi	Cannot convert parameter 1 from 'float' to 'int': Không thể chuyển đổi tham số thứ nhất từ kiểu 'float' sang kiểu '&int'
6	define (v) definition (n)	định nghĩa	Entry point must be defined: Hàm <i>main()</i> đã được định nghĩa
7	entry point	điểm vào của chương trình (hàm <i>main()</i>)	
8	exception	ngoại lệ	Unhandled exception: Ngoại lệ không điều khiển được
9	expected	chờ đợi, mong đợi	Expected constant expression: Chờ đợi biểu thức hằng
10	expression	biểu thức	Constant expression: Biểu thức hằng
11	find/found	tìm thấy	Identifier not found: Định danh không tìm thấy
12	forget	quên	Did you forget a ';': Bạn quên dấu ';'?
13	follow	theo sau	'x' followed by 'void' is illegal
14	function	hàm	'void' function : hàm 'void'
15	identifier	định danh (biến, hàm, ...)	Identifier not found: Định danh không tìm thấy
16	illegal	không hợp lệ	Illegal else without matching if: <i>else</i> không ứng với <i>if</i> thì không hợp lệ
17	initialize	khởi tạo	The variable 'r' being used without being initialized: Biến 'r' đang được sử

			<i>dùng mà chưa khởi tạo (giá trị)</i>
18	local	<i>cục bộ</i>	Local function definitions are illegal: <i>Định nghĩa hàm cục bộ là không hợp lệ</i>
19	matching	<i>tương ứng</i>	Illegal else without matching if: <i>else không ứng với if thì không hợp lệ</i>
20	member	<i>thành viên, thành phần (của struct/class/union)</i>	'x' is not a member of struct : <i>'x' không phải là thành phần của struct</i>
22	missing	<i>thiếu</i>	Missing ': ' before 'if': <i>Thiếu dấu ';' trước 'if'</i>
23	parameter	<i>tham số</i>	Cannot convert parameter 1 from 'float' to 'int&': <i>Không thể chuyển đổi tham số thứ nhất từ kiểu 'float' sang kiểu '&int'</i>
24	redefinition	<i>định nghĩa lại</i>	'int i' : redefinition : <i>Định nghĩa lại biến i</i>
25	return	<i>trả về</i>	...must be return a value: <i>...phải trả về giá trị</i>
26	undeclared	<i>chưa khai báo</i>	Undeclared identifier: <i>Định danh chưa khai báo</i>
27	unnhandled	<i>Không điều khiển/kiểm soát được</i>	Unhandled exception: <i>Ngoại lệ không điều khiển được (ngoài tầm kiểm soát)</i>
28	value	<i>giá trị</i>	...must be return a value: <i>...phải trả về giá trị</i>
29	variable	<i>biến</i>	The variable 'r' being used without being initialized: <i>Biến 'r' đang được sử dụng mà chưa khởi tạo (giá trị)</i>
30	violation	<i>sự vi phạm</i>	Access violation at address...: <i>Vi phạm truy cập tại địa chỉ...</i>

MỤC LỤC

LỜI NÓI ĐẦU	1
PHẦN 1: TỔNG QUAN VỀ CÁC LOẠI LỖI	2
PHẦN 2: CÁCH SỬA CÁC LOẠI LỖI	3
I. CÁCH SỬA LỖI CÚ PHÁP	3
1. Cách xem lỗi cú pháp.....	3
2. Cách sửa lỗi cú pháp	3
3. Các câu thông báo lỗi cú pháp thường gặp.....	4
4. Một số minh họa về cách sửa lỗi cú pháp	5
II. CÁCH SỬA LỖI THỜI GIAN CHẠY	16
1. Cách sửa lỗi thời gian chạy.....	17
2. Các ví dụ minh họa về lỗi thời gian chạy	17
III. CÁCH SỬA LỖI LOGIC	17
1. Chạy từng lệnh.....	18
2. Dừng chương trình tại các điểm dừng	19
3. Xem giá trị các biến khi debug	22
4. Ví dụ tìm lỗi logic bằng công cụ Debug.....	23
PHẦN 3:MỘT SỐ TỪ/CỤM TỪ TIẾNG ANH THƯỜNG GẶP TRONG CÁC CÂU THÔNG BÁO LỖI.....	28

