# Question 2: Variable

**Q2.1: Use a nest 3-for-loops, calculate the time to run when use, and not use register. Test in several times, and give a statistics when MAX is 10, 100, 1000, 10000.**

|  |  |  |
| --- | --- | --- |
|  | Non-register | Register |
| 10 | 0 ms | 0 ms |
| 100 | 3 - 5 ms | 2 - 5 ms |
| 1000 | 2300 – 2400 ms | 2300 – 2400 ms |
| 10000 | Quá lâu > 8p chưa chạy xong :3 |  |

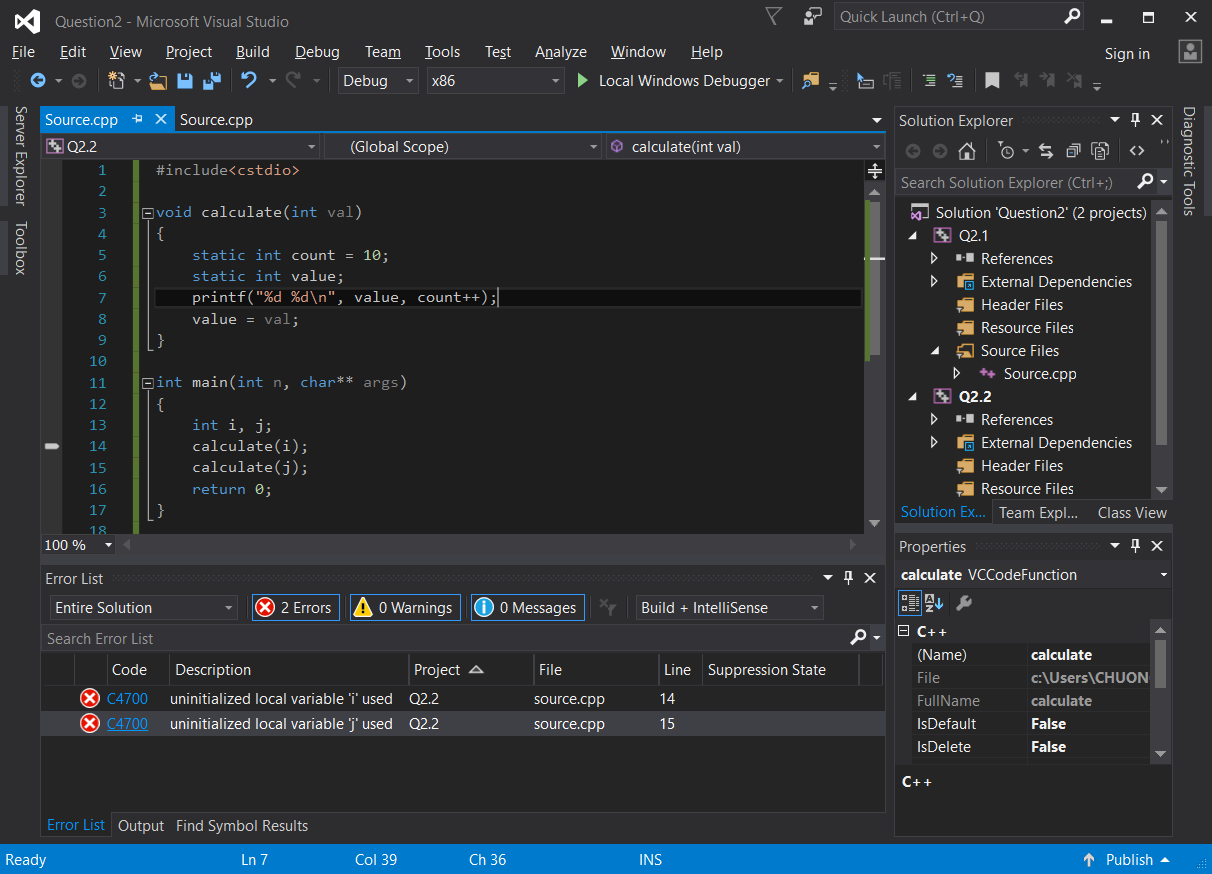
Có thể do VS đã tối ưu hóa mã nguồn trước khi dịch sang mã máy nên 2 kết quả tương đồng nhau.

**Q2.2: Implement the following**

|  |
| --- |
| void calculate(int val)  {  static int count=10;  static int value;  printf("%d %d\n",value, count++);  value = val;  }  int main(int n, char\*\* args)  {  int i, j;  calculate(i);  calculate(j);  } |

Look at the result and explain why?

Biên dịch báo lỗi hai biến i,j chưa được khởi tạo giá trị trước khi sử dụng.



Nếu có thể thực hiện được thì kết quả sẽ là :

0 10

? 11

* Lần gọi hàm calculate thứ nhất: in ra giá trị biến tĩnh value chưa được khởi tạo giá trị nên có giá trị mặc định là 0. Sau đó biến tĩnh value được gán giá trị bằng val (có giá trị truyền vào bởi biến i). Mà hai biến i,j (biến cục bô) chưa được khởi tạo giá trị nên sẽ có giá trị ngẫu nhiên.
* Lần 2: như đã nêu trên, i nhận giá trị ngẫu nhiên nên in ra giá trị value cũng ngẫu nhiên. Biến count là biến tĩnh nên chỉ được khai báo 1 lần và tồn tại cho đến khi kết thúc chương trình. Sau lần gọi thứ nhất, biến count được tăng lên 1 đến lần gọi hàm Calculate thứ 2 giá trị của nó là 11.

**Q2.3: For counting number of times a function has been called, we use three ways:**

* **Global variable**
* **Static variable inside function**
* **Static variable outside function**

**Implements those ways, and compare.**

**Code:**

#include<iostream>

using namespace std;

int count\_no\_function;

//su dung bien toan cuc

void global\_variable() {

count\_no\_function++;

}

// su dung bien tinh cuc bo

int static\_variable\_inside() {

static int sta\_va\_in;

return ++sta\_va\_in;

}

// su dung bien tinh o ngoai ham

void static\_variable\_outside(int &a) {

a++;

}

int main() {

static int sta\_va\_out;

int sta\_va\_in = 0;

cout << "so lan goi ham global\_variable : " << count\_no\_function << endl;

cout << "so lan goi ham static\_variable\_inside : " << sta\_va\_in << endl;

cout << "so lan goi ham static\_variable\_outside : " << sta\_va\_out << endl;

global\_variable();

sta\_va\_in=static\_variable\_inside();

static\_variable\_outside(sta\_va\_out);

cout << "so lan goi ham global\_variable : " << count\_no\_function << endl;

cout << "so lan goi ham static\_variable\_inside : " << sta\_va\_in << endl;

cout << "so lan goi ham static\_variable\_outside : " << sta\_va\_out << endl;

return 0;

}

**So sánh:**

Khi sử dụng biến toàn cục: hàm không cần trả về giá trị hay truyền tham số vào hàm. Các hàm đều có thể truy cập vào biến toàn cục để lấy giá trị của nó.

Khi sử dụng biến cụ bộ tĩnh ở trong hàm: hàm cần phải trả về giá trị để có thể in được số lần gọi hàm trong hàm main().

Khi sử dụng biến tĩnh nằm ngoài hàm:

* Nếu là biến tĩnh toàn cục thì như biến tự động toàn cục.
* Nếu là biến tĩnh cục bộ trong hàm main() thì cần phải truyền tham số cho hàm để có thể thay đổi giá trị (đếm số lần gọi hàm).

**Q2.4: A programmer codes his program with 2 files: main.cpp and another.cpp. In another.cpp, he declares a global variable named myVar with initial value is 10. In main .cpp, he needs to print value of myVar into screen. And he tries the following ways:**

* **Declare myVar in another.cpp, and in print it in main() in main.cpp**
* **Declare myVar in another.cpp. Declare again myVar in main.cpp and print it**
* **Declare myVar in another.cpp. In main.cpp, declare myVar with extern keyword, and print it.**

**He’s got problem with the first and second way. The third is run well. Please explain what happen with the first and second; explain the reason why the third is OK and try to implement it.**

* Cách 1: Chương trình dịch sẽ báo lỗi biến myVar chưa được khai báo (biến myVar chưa được khái báo trong hàm main).
* Cách 2: Chương trình dịch thành công, nhưng khi chạy thực hiện thì không in ra giá trị biến myVar = 10 mà là một giá trị ngẫu nhiên (biến myVar trong hàm main và biến myVar trong file another.cpp không liên quan nhau).
* Cách 3: Chương trình chạy ok (biến myVar trong hàm main được khai báo với từ khóa extern cho phép biến được khái báo trong một file .cpp khác - ở đây là file another.cpp)
* Thực hiện:

File another.cpp

int myVar = 10;

File source.cpp

#include<iostream>

using namespace std;

int main() {

extern int myVar;

cout <<"myVar = "<< myVar << endl;

return 0;

}

**Q2.5: Detect & solve problem of the following code:**

|  |
| --- |
| #include <stdio.h>  const int xyz;  int main(int n, char\*\* args)  {  xyz = 0;  printf("%d", xyz);  } |

const int xyz; >< xyz = 0;

xyz là hằng số không thể thay đổi giá trị.

**Giải pháp 1:**

|  |
| --- |
| #include <stdio.h>  int xyz;  int main(int n, char\*\* args)  {  xyz = 0;  printf("%d", xyz);  } |

**Giải pháp 2:**

|  |
| --- |
| #include <stdio.h>  const int xyz;  int main(int n, char\*\* args)  {  printf("%d", xyz);  } |