**Sự khác nhau giữa Macro, Inline và Function**

1. Macro:

* Được xử lý bởi preprocessor
* Thay thế đoạn code được khai báo macro vào bất cứ chỗ nào xuất hiện macro đó
* VD: #define SUM(a,b)     (a+b)
  + Preprocessor khi gặp bất kỳ lời gọi SUM(first+last) nào thì thay ngay bằng (first+last)

2. Inline

* Được xử lý bởi compiler
* Được khai báo với từ khóa inline
* Khi compiler thấy bất kỳ chỗ nào xuất hiện inline function, nó sẽ thay thế chỗ đó bởi định nghĩa của hàm đã được compile tương ứng. –> Phần được thay thế không phải code mà là đoạn code đã được compile

3. Hàm bình thường

* Khi thấy hàm được gọi, compiler sẽ phải lưu con trỏ chương trình PC hiện tại vào stack; chuyển PC tới hàm được gọi, thực hiện hàm đó xong và lấy kết quả trả về; sau đó quay lại vị trí ban đầu trong stack trước khi gọi hàm và tiếp tục thực hiện chương trình.
* Như có thể thấy, các này khiến chương trình tốn thời gian hơn là chỉ cần thay thế đoạn code đã được compile (cách của inline function)

4. So sánh

* Macro đơn giản là chỉ thay thế đoạn code macro vào chỗ được gọi trước khi được biên dịch
* Inline thay thế đoạn mã code đã được biên dịch vào chỗ được gọi
* Hàm bình thường phải tạo một function call, lưu địa chỉ trước khi gọi hàm vào stack sau đó mới thực hiện hàm và sau cùng là quay trở về địa chỉ trên stack trước khi gọi hàm và thực hiện tiếp chương trình
* Macro khiến code trở nên dài hơn rất nhiều so với bình thường nhưng thời gian chạy nhanh.
* Hàm inline cũng khiến code dài hơn, tuy nhiên nó làm giảm thời gian chạy chương trình
* Hàm bình thường sẽ phải gọi function call nên tốn thời gian hơn inline function nhưng code ngắn gọn hơn.