1. Khi nhấp vào nút "sync":
    - chương trình bắt đầu bằng việc xóa nội dung trong div "output", sau đó in ra chuỗi "BEGIN sync". Tiếp theo, hàm "sFunction" được gọi và trả về chuỗi "Hello S!". Cuối cùng, chuỗi "Hello S!" và chuỗi "END." được in ra trong div "output".
    - Call Stack: `syncFunction()`
    - Macrotask Queue: Trống
    - Microtask Queue: Trống
2. Khi nhấp vào nút "settimeout":
    - chương trình cũng bắt đầu bằng việc xóa nội dung trong div "output" và in ra chuỗi "BEGIN settimeout". Tiếp theo, một hàm callback được truyền vào hàm "setTimeout", và được lên lịch để thực thi sau 0ms. Sau đó, chuỗi "END." được in ra trong div "output". Sau khi toàn bộ mã nguồn trong hàm hiện tại được thực thi xong, hàm callback được gọi và chuỗi "Hello S!" được in ra.
    - Call Stack: `setTimeout(callback, 0)`
    - Macrotask Queue: `callback`
    - Microtask Queue: Trống
3. Khi nhấp vào nút "promise":
    - chương trình tiếp tục bằng việc xóa nội dung trong div "output" và in ra chuỗi "BEGIN promise". Hàm "aFunction1" được gọi, và bên trong hàm này, một Promise được tạo và trả về ngay lập tức với giá trị "Hello A1!". Tiếp theo, hàm "then" được gọi trên Promise trả về, và một hàm callback được truyền vào. Trong hàm callback này, chuỗi "Hello A1!" được in ra trong div "output". Cuối cùng, chuỗi "END." được in ra.
    - Call Stack: `aFunction1()`
    - Macrotask Queue: Trống
    - Microtask Queue: Trống
4. Khi nhấp vào nút "async":

- chương trình lại bắt đầu bằng việc xóa nội dung trong div "output" và in ra chuỗi "BEGIN async". Hàm "aFunction2" được gọi, và bên trong hàm này, một Promise được tạo và trả về ngay lập tức với giá trị "Hello A2!". Tiếp theo, hàm "then" được gọi trên Promise trả về và một hàm callback được truyền vào. Trong hàm callback này, chuỗi "Hello A2!" được in ra trong div "output". Cuối cùng, chuỗi "END." được in ra.
- Call Stack: `aFunction2()`
- Macrotask Queue: Trống
- Microtask Queue: Trống

5. Khi nhấp vào nút "await":
    - chương trình xóa nội dung trong div "output" và in ra chuỗi "BEGIN await". Tiếp theo, hàm "aFunction1" được gọi và trả về một Promise. Trong dòng tiếp theo, hàm "await" được sử dụng để chờ Promise hoàn thành và trả về kết quả. Khi Promise hoàn thành, giá trị "Hello A1!" được gán cho biến "ret". Sau đó, chuỗi "Hello A1!" và chuỗi "END." được in ra trong div "output".
    - Call Stack: `aFunction1()`
    - Macrotask Queue: Trống
    - Microtask Queue: Trống
6. Khi nhấp vào nút "async vs sync":
    - chương trình bắt đầu bằng việc xóa nội dung trong div "output" và in ra chuỗi "BEGIN sync vs async". Tiếp theo, hàm "aFunction1" được gọi và trả về một Promise. Hàm "then" được gọi trên Promise này và một hàm callback được truyền vào, trong đó chuỗi "Hello A1!" được in ra trong div "output". Sau đó, hàm "aFunction2" được gọi và trả về một Promise khác. Hàm "then" được gọi trên Promise này và một hàm callback được truyền vào, trong đó chuỗi "Hello A2!" được in ra trong div "output". Cuối cùng, hàm "sFunction" được gọi và chuỗi "Hello S!"được trả về và in ra trong div "output". Cuối cùng, chuỗi "END." được in ra trong div "output".
    - Call Stack: `aFunction1()`
    - Macrotask Queue: Trống

- Microtask Queue: Trống

7. Khi nhấp vào nút "waitall":
    - Call Stack: `waitAll()`
    - Macrotask Queue: Trống
    - Microtask Queue: Trống
    - Chương trình bắt đầu bằng việc gọi hàm `waitAll()`.
    - Trong hàm này, có ba hàm con được gọi lần lượt: `aFunction1()`, `aFunction2()` và `sFunction()`.
    - Mỗi hàm con trả về một Promise với các giá trị tương ứng là "Hello A1!", "Hello A2!" và "Hello S!".
    - Hàm `Promise.all()` được gọi và truyền vào một mảng chứa các Promise trên.
    - Promise.all() trả về một Promise mới, chờ đợi tất cả các Promise trong mảng hoàn thành.
    - Khi tất cả các Promise hoàn thành, một mảng kết quả được trả về chứa các giá trị của các Promise tương ứng.
    - Kết quả cuối cùng là một mảng ["Hello A1!", "Hello A2!", "Hello S!"].
    - Mảng này được in ra trên giao diện người dùng.
8. Khi nhấp vào nút "mixed":
    - Call stack: Ban đầu rỗng. Sau khi nhấn nút, các hàm được gọi từ sự kiện click sẽ được thêm vào call stack và thực thi, sau đó bị xóa khỏi stack.
    - Macrotask queue: Hai hàm setTimeout được thêm vào hàng đợi macrotask với thời gian chờ là 10ms và 0ms.
    - Microtask queue: Không có microtask nào được thêm vào hàng đợi microtask trong quá trình thực thi này.