**GIẢI THÍCH**

**Bao gồm các bước thực hiện:**

**1.Nhập dữ liệu**

**Người dùng sẽ nhập số lượng khách hàng và thông tin về thời gian bắt đầu (ai), thời gian kết thúc (bi), và lợi nhuận (vi) cho mỗi khách hàng.**

**2. Sắp Xếp Dữ Liệu:**

**Các cuộc hẹn (hoặc khoảng thời gian thuê máy) sẽ được sắp xếp theo thời gian kết thúc (bi). Điều này giúp chúng ta dễ dàng xác định các khoảng thời gian không giao nhau.**

**3. Xây Dựng Bảng Quy Hoạch Động:**

**Sử dụng một mảng f để lưu trữ lợi nhuận tối đa có thể thu được cho mỗi khách hàng.**

**Với mỗi khách hàng, chúng ta sẽ xem xét hai lựa chọn:**

**Lựa chọn 1: Không phục vụ khách hàng này (lợi nhuận là lợi nhuận tối đa của khách hàng trước đó).**

**Lựa chọn 2: Phục vụ khách hàng này, cộng với lợi nhuận từ khách hàng không giao nhau trước đó.**

**4 Tìm Lợi Nhuận Tối Đa:**

**Cuối cùng, giá trị lớn nhất trong mảng f sẽ là lợi nhuận tối đa có thể đạt được.**

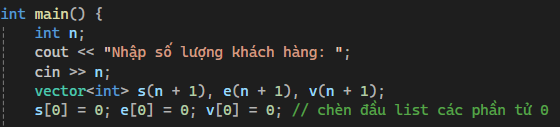
**Giải thích thuật toán**

****

**#include <iostream>: Thư viện cần thiết để sử dụng các chức năng nhập xuất như cin và cout.**

**#include <vector>: Cho phép sử dụng cấu trúc dữ liệu vector để lưu trữ các giá trị động.**

**#include <algorithm>: Cung cấp các hàm hỗ trợ như sort và max\_element.**

****

**int n;: Khai báo biến n để lưu số lượng khách hàng.**

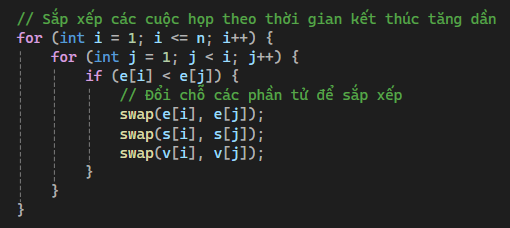
**vector<int> s(n + 1), e(n + 1), v(n + 1);: Khai báo ba vector:**

**s: Lưu thời gian bắt đầu cho từng khách hàng.**

**e: Lưu thời gian kết thúc cho từng khách hàng.**

**v: Lưu lợi nhuận tương ứng với từng khách hàng.**

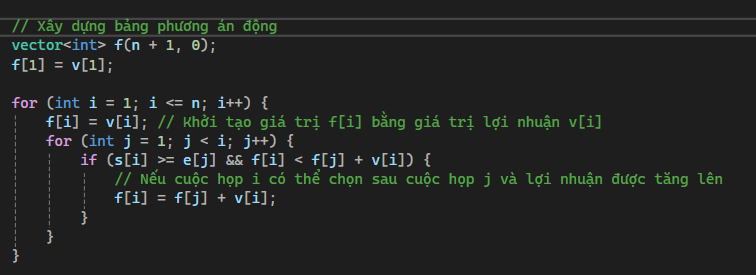
**s[0] = 0; e[0] = 0; v[0] = 0;: Khởi tạo phần tử đầu tiên của mỗi vector với giá trị 0 để dễ dàng xử lý trong thuật toán.**

****

**Vòng lặp for từ 1 đến n yêu cầu người dùng nhập thời gian bắt đầu, thời gian kết thúc và lợi nhuận cho từng khách hàng.**

**Sắp Xếp:**

* **Hai vòng lặp lồng nhau để sắp xếp các cuộc họp theo thời gian kết thúc tăng dần. Nếu thời gian kết thúc của cuộc họp i nhỏ hơn thời gian kết thúc của cuộc họp j, nó sẽ hoán đổi thông tin giữa chúng.**
* **Sử dụng swap để hoán đổi giá trị giữa các vector.**

****

**Khởi tạo giá trị:**

* **f[i] là một mảng dùng để lưu lợi nhuận tối đa đạt được khi chọn đến cuộc họp thứ i.**
* **t[i] là mảng dùng để lưu chỉ số của cuộc họp trước đó mà có thể chọn để đạt lợi nhuận tối đa, giúp truy vết lại các cuộc họp được chọn.**
* **v[i] là mảng lợi nhuận của từng cuộc họp.**
* **s[i] là thời gian bắt đầu của cuộc họp thứ i.**
* **e[i] là thời gian kết thúc của cuộc họp thứ i.**

**Duyệt qua các cuộc họp:**

* **for (int i = 1; i <= n; i++): Vòng lặp này duyệt qua tất cả các cuộc họp từ 1 đến n.**
* **f[i] = v[i];: Khởi tạo lợi nhuận ban đầu cho cuộc họp thứ i là giá trị lợi nhuận của nó (v[i]), giả định rằng nếu chỉ chọn một mình cuộc họp i, thì lợi nhuận đạt được là v[i].**

**So sánh với các cuộc họp trước đó:**

* **for (int j = 1; j < i; j++): Vòng lặp lồng trong để duyệt qua tất cả các cuộc họp j trước i (từ 1 đến i-1).**
* **if (s[i] >= e[j] && f[i] < f[j] + v[i]): Kiểm tra xem cuộc họp i có thể chọn được sau cuộc họp j (nghĩa là s[i] >= e[j]) và liệu lợi nhuận thu được khi chọn cuộc họp i sau j có lớn hơn lợi nhuận hiện tại của f[i] hay không.**
  + **Nếu điều kiện này thỏa mãn, thì cập nhật lợi nhuận tối đa cho f[i] bằng cách thêm lợi nhuận v[i] vào lợi nhuận tối đa của f[j].**

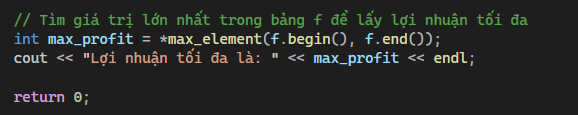
**Vd minh họa:**

****

**Áp dụng đoạn mã:**

* **Khởi tạo:**
  + **f[1] = v[1] = 5**
  + **f[2] = v[2] = 6**
  + **f[3] = v[3] = 5**
  + **f[4] = v[4] = 8**
* **Duyệt từng cuộc họp:**
  + **Cuộc họp 2:**
    - **Không thể chọn cuộc họp 2 sau cuộc họp 1 vì s[2] < e[1].**
  + **Cuộc họp 3:**
    - **So với cuộc họp 1: s[3] >= e[1] và f[3] < f[1] + v[3], nên f[3] = f[1] + v[3] = 5 + 5 = 10, và t[3] = 1.**
    - **So với cuộc họp 2: không thỏa mãn điều kiện.**
  + **Cuộc họp 4:**
    - **So với cuộc họp 1: s[4] >= e[1] và f[4] < f[1] + v[4], nên f[4] = f[1] + v[4] = 5 + 8 = 13, và t[4] = 1.**
    - **So với cuộc họp 3: s[4] >= e[3] và f[4] < f[3] + v[4], nên f[4] = f[3] + v[4] = 10 + 8 = 18, và t[4] = 3.**

**Kết quả, lợi nhuận tối đa là 18, và các cuộc họp được chọn là cuộc họp 1, 3 và 4.**

****

**Tìm lợi nhuận và in ra kết quả**

**( không hiểu nữa thì chịu ) >>>>>>>>>**