|  |  |  |
| --- | --- | --- |
|  | Trung ương hội Olympiad Tin học Châu Âu năm 2007  Ngày 1 - 7 | VÒNG CỔ  Trang 1/5 |

**VÒNG CỔ**

|  |  |
| --- | --- |
| ***File nguồn :***  ***Thời gian giới hạn:***  ***Giới hạn bộ nhớ:*** | necklace.c/necklace.cpp/necklace.pas  3 s  128 MB |

Alice nói: “Tớ có một chiếc vòng cổ đẹp nhất trên đời này.Từ trái sang phải, nó có hai viên ngọc trai màu đỏ, hai viên màu xanh lá cây và thêm một viên màu đỏ nữa”. Beatrix nhanh chóng tiếp lời: “Của tớ còn tốt hơn. Nó gần giống như chiếc vòng của cậu nhưng cậu cần phải thay hai viên ngọc trai ngoài cùng bên phải và thay thế chúng bằng hai viên màu xanh dương”. Cô vừa dứt lời thì Caroline nhảy vào: “Thế chưa là gì so với vòng cổ của tớ. Của tớ còn có thêm một viên màu vàng bên trái nữa”. Bạn chắc cũng sẽ chẳng ngạc nhiên khi Dominica cũng chẳng chịu ngồi yên: “Tất cả đều chán phèo. Để có được chiếc vòng cổ như của tớ, cậu phải lấy chiếc vòng cổ của Beatrice, bỏ viên ngọc trai ngoài cùng bên trái và bên phải đi rồi thêm hai viên ngọc trai màu đen vào bên trái”. Và câu chuyện cứ tiếp diễn cho đến khi Zaida hỏi: “Tớ hơi bối rối một chút, nhưng viên ngọc trai ngoài cùng bên trái vòng cổ của Eugenie có màu gì?”.

**CÔNG VIỆC**

Nhiệm vụ của bạn là tạo một thư viện (thống nhất là bằng Pascal) để giúp các cuộc hội thoại kiểu này. Giao diện của thư viện sẽ được miêu tả trong phần tiếp theo; bạn có thể tìm thấy các thư viện mẫu trong các thư mục sau /mo/public/necklace/c, /mo/public/necklace/cpp, và/mo/public/necklace/pas (các thư mục con trong C và C++ cũng chứa các tệp tin tiêu đề necklace.h). Không có dữ liệu đầu vào hay dữ liệu đầu ra.

Thư viện của bạn sẽ làm việc với một tập hợp các day chuyền. Mỗi một chiếc vòng cổ là một chuỗi số nguyên lớn hơn 0 và nhỏ hơn 1 000 000, theo thứ tự từ trái sang phải. Mỗi một chiếc vòng cổ được xác định bởi một số nguyên không âm. Ban đầu, chỉ có chiếc vòng cổ mang thứ tự 0 là một chuỗi rỗng tượng trưng cho hiện tại.

Ban đầu, hệ thống đánh giá sẽ gọi thủ tục ***init*** một lần duy nhất. Sau đó, nó sẽ lien tục gọi theo một trật tự tuỳ ý các hàm ***create*** và ***pearl***. Tổng cộng, thư viện của bạn sẽ chạy ít nhất là 1 000 000 lần mỗi lần chạy thử.

|  |  |  |
| --- | --- | --- |
|  | Trung ương hội Olympiad Tin học Châu Âu năm 2007  Ngày 1 - 7 | VÒNG CỔ  Trang 2/5 |

Hàm ***create*** mỗi lần được gọi sẽ tạo ra một chiếc vòng cổ tương ứng với mỗi lần gọi. Số lượng vòng cổ sẽ nhiều hơn 1 đơn vị số lượng vòng cổ lớn nhất hiện có, tức là lần đầu tiên gọi hàm ***create*** tạo ra chiếc vòng mang số 1, lần gọi thứ hai tạo ra chiếc vòng mang số 2, v.v… Chiếc vòng mới sẽ được tạo nên dựa trên cơ sở của chiếc vòng ***from*** nhờ các hàm quy định bởi các thông số ***operation, on\_left*** và ***param***:

* Nếu tham số ***operation***  là chữ R (trong “remove” – loại bỏ) thì một số nguyên ở cuối chiếc vòng sẽ bị loại bỏ. Hàm ***param*** sẽ bị bỏ qua trong trường hợp này.
* Nếu tham số ***operation*** là chữ A (trong “add” – thêm) thì số nguyên ***param*** sẽ được thêm vào cuối chiếc vòng.

Nếu giá trị của ***on\_left*** là đúng (true) (trong C là khác 0) hàm sẽ thực hiện công việc của mình ở ngoài cùng bên trái của chiếc vòng cổ, ngược lại nó sẽ thực hiện công việc ở ngoài cùng bên phải của chiếc vòng cổ. Thông số ***from*** của hàm ***create*** được gọi luôn nhỏ hơn số thứ tự của chiếc vòng tạo nên bởi hàm lúc này, nghĩa là nó được tạo nên dựa trên một chiếc vòng trước đó. Bạn có thể giả định rằng sẽ không có bất cứ trường hợp nào yêu cầu loại bỏ một viên ngọc trai từ một dây chuyên rỗng.

Hàm ***pearl*** sẽ trả về giá trị của viên ngọc trai ngoài cùng bên trái nếu như ***on\_left*** trả về giá trị đúng (true) (khác 0 đối với C/C++), bằng không, nó sẽ trả về giá trị của viên ngọc trai ngoài cùng bên phải của chiếc vòng **neck\_id**. Hàm ***pearl*** sẽ chỉ đượ gọi bởi một chiếc vong cổ không rỗng đã được tạo ra bởi lần gọi hàm  ***create*** trước đó. Hàm này sẽ không làm thay đổi chiếc vòng cổ.

**MÔ TẢ GIAO DIỆN**

**Trong C/C++:**

*extern void init (void);*

*extern void create (int from, char operation, int on\_left, int param);*

*extern int pearl (int neck\_id, int on\_left);*

**Trong Pascal:**

*unit necklace;*

*interface*

*procedure init;*

*procedure create (from : longint; operation : char; on\_left : boolean; param : longint);*

*function pearl (neck\_id : longint; on\_left : boolean) : longint;*

|  |  |  |
| --- | --- | --- |
|  | Trung ương hội Olympiad Tin học Châu Âu năm 2007  Ngày 1 - 7 | VÒNG CỔ  Trang 3/5 |

Lưu ý rằng bạn có thể tạo các thủ tục bổ sung, chức năng, và các biến toàn cầu trong thư viện bạn chuẩn bị.

**VÍ DỤ**

Ví dụ sau đây cho thấy một trường hợp có thể xảy ra sau nhiều lần gọi hàm và giá trị trả về:

**Trong C/C++:**

*init ();*

*create (0, ’A’, 1, 5);*

*create (1, ’A’, 1, 3);*

*pearl (2, 0); /\* returns 5 \*/*

*create (2, ’R’, 0, 0);*

*pearl (3, 0); /\* returns 3 \*/*

*pearl (2, 0); /\* returns 5 \*/*

**Trong Pascal:**

*init;*

*create (0, ’A’, true, 5);*

*create (1, ’A’, true, 3);*

*pearl (2, false); { returns 5 }*

*create (2, ’R’, false, 0);*

*pearl (3, false); { returns 3 }*

*pearl (2, false); { returns 5 }*

**KIỂM TRA THƯ VIỆN CỦA BẠN**

Ngoài các thư viện mẫu bạn của bạn, thư mục con /mo/public/necklace chứa các file neck\_main.c, neck\_main.cpp, và neck\_main.pas là các file nguồn chính có thể biên dịch được với thư viện của bạn và sẽ kiểm tra những thư viện thông thường theo mẫu trên (chương trình sẽ cho bạn biết giá trị trả về của hàm ***pearl*** có chính xác hay không). Chương trình **compile** với tham số **necklace** sẽ biên dịch thư viện của bạn bằng các file nguồn chính (neck\_main.c, neck\_main.cpp hoặc neck\_main.pas) và kết quả trả về của chương trình **necklace** sẽ cho bạn kiểm tra thư viện của mình như mô tả trên1. Nếu chương trình chạy mà không có tham số, nó sẽ gọi thư viện của bạn như ví dụ và kiểm tra xem kết quả trả về của hàm ***pearl*** có chính xác hay không (nếu không chính xác, một tin nhắn sẽ được viết ở đầu ra

1 Lưu ý rằng bạn có thể chạy chương trình **compile** với một tệp tin như một tham số điều sẽ cho phép bạn biên dịch thư viện với các tệp tin chính của riêng bạn

|  |  |  |
| --- | --- | --- |
|  | Trung ương hội Olympiad Tin học Châu Âu năm 2007  Ngày 1 - 7 | VÒNG CỔ  Trang 4/5 |

sai số chuẩn). Nếu chương trình được gọi với tham số *-i­* thì chương trình sẽ chuyển

sang chế độ tương tác. Chương trình sẽ đọc dữ liệu đầu vào và cho ra dữ liệu đầu ra tiêu chuẩn.

Trong chế độ tương tác, chương trình đầu tiên sẽ gọi thủ tục ***init*** và sau khi thủ tục này kết thúc, nó sẽ cho ra kết quả ***Initiated***. Sau đó, ba loại lệnh sau đây được nhập vào:

* **create *from operation on\_left param***

Chương trình khi thấy lệnh này sẽ gọi hàm ***create*** với các tham số ***from, operatio, on\_left, param***. Sau khi hàm kết thúc, chương trình xuất ra “**Done**”.

* **pearl *neck\_ ip on\_left***

Chương trình khi thấy lệnh này sẽ gọi hàm ***pearl*** với các tham số ***neck\_ id, on\_left***. Sau khi hàm kết thúc, chương trình xuất ra số “ Return X” với X là giá trị trả về của hàm .

* **quit**

Câu lệnh này sẽ kết thúc chương trình.

Lưu ý rằng các chương trình trong chế độ tương tác chỉ cung cấp cho bạn với câu trả lời của thư viện của bạn và nósẽ không kiểm tra xem các câu trả lời là đúng hay sai.

Sau đây là những câu lệnh và kết quả trả về khi chương trình đang ở chế độ tương tác tương ứng với những lệnh đến thư viện từ ví dụ:

**Trong C/C++:**

*Initiated*

*create 0 A 1 5*

*Done*

*create 1 A 1 3*

*Done*

*pearl 2 0*

*Returns 5*

*create 2 R 0 0*

*Done*

*pearl 3 0*

*Returns 3*

*pearl 2 0*

*Returns 5*

*quit*

|  |  |  |
| --- | --- | --- |
|  | Trung ương hội Olympiad Tin học Châu Âu năm 2007  Ngày 1 - 7 | VÒNG CỔ  Trang 5/5 |

**Trong Pascal:**

*Initiated*

*create 0 A true 5*

*Done*

*create 1 A true 3*

*Done*

*pearl 2 false*

*Returns 5*

*create 2 R false 0*

*Done*

*pearl 3 false*

*Returns 3*

*pearl 2 false*

*Returns 5*

*quit*