

Postfix Evaluation

จงเขียนโปรแกรมที่ทำการคำนวณค่าของนิพจน์แบบ post fix โดยให้เขียนฟังก์ชัน `int eval_postfix(vector<pair<int,int>> v)` ซึ่งนิพจน์แบบ post fix นั้นจะเก็บไว้ใน vector `v` เรียงจากซ้ายไปขวาในช่อง `v[0]` ถึง `v[v.size()-1]` ตามลำดับ

นิพจน์แบบ postfix จะเก็บใน `v` ในรูปแบบดังต่อไปนี้ `v[i].first` นั้นจะมีค่าเป็น 0 หรือ 1 เท่านั้นเพื่อระบุว่า term ดังกล่าวเป็น operator หรือ operand โดยที่ `v[i].first` มีค่าเป็น 0 แสดงว่า เป็น operator ส่วนค่า `v[i].second` นั้น ในกรณีที่ `v[i].first` เป็น 0 จะระบุ operator โดยที่ `v[i].second` จะมีค่าตั้งแต่ 0 ถึง 3 ซึ่งระบุถึง operator บวก, ลบ, คูณ,หารไม่เอาเศษ ตามลำดับ แต่เมื่อ `v[i].first` มีค่าเป็น 1 `v[i].second` จะระบุถึง operand เป็น int ซึ่งมีค่าอยู่ในช่วง 0 ถึง 999

ข้อบังคับ

โจทย์จะมีไฟล์เริ่มต้น `main.cpp` และ `student.h` ให้ และให้นักศึกษาเขียน code ในไฟล์ `student.h` เท่านั้น (ในไฟล์จะมีฟังก์ชัน `eval_postfix` ให้อยู่แล้ว) นักศึกษาสามารถใช้โครงสร้างข้อมูลใด ๆ ของ stl ได้โดยไม่มีข้อกำหนด และสามารถ `#include` เพิ่มเติมใน `student.h` ได้อย่างอิสระ

ข้อกำหนดขนาดข้อมูล

รับประกันว่าขนาดของข้อมูลใน vector จะไม่เกิน 10,000 ตัว และเป็น postfix ที่อยู่ในรูปแบบที่ถูกต้อง

ตัวอย่าง

vector a	Postfix	ผลลัพธ์
<code><(1,10)></code>	<code>+ 10</code>	10
<code><(1,10), (1,15), (0,0)></code>	<code>10 15 +</code>	25
<code><(1,10), (1,15), (0,1)></code>	<code>10 15 -</code>	-5
<code><(1,9), (1,3), (0,3), (1,2), (0,2)></code>	<code>9 3 / 2 *</code>	6
<code><(1,7), (1,3), (0,3), (1,2), (0,2)></code>	<code>7 3 / 2 *</code>	4
<code><(1,1), (1,2), (1,3), (1,4), (0,0), (0,0), (0,0)></code>	<code>1 2 3 4 + + +</code>	10