
Reducció de programes

X18215_ca

Cada cas d'entrada d'aquest exercici és un string sobre l'alfabet $\{+, *, =, <, \{, \}, (,), ;, v, n, i, e, w\}$. Per exemple, aquest seria un possible string d'entrada: `"v=n+n*v;i(v<n){v=v+n;}ew(n<v*v)v=n+v;"`. Aquestes entrades tenen una pinta una mica críptica. A continuació donem una petita explicació del seu significat i el sentit de l'exercici. Remarquem, però, que aquesta explicació no és imprescindible per a poder resoldre l'exercici. La podeu obviar si voleu.

Explicació (la podeu ometre)

Aquests strings d'entrada són una manera simplificada de descriure la seqüència de tokens d'un programa en un cert llenguatge de programació. Per a tenir una visió més intuïtiva, podeu interpretar cada lletra com segueix:

- v: variable
- n: number
- i: if
- e: else
- w: while

Per exemple, l'string anterior seria la versió simplificada d'aquesta seqüència de tokens:

```
variable = number + number * variable ;
if ( variable < number ) {
    variable = variable + number ;
} else
    while ( number < variable * variable )
        variable = number + variable ;
```

Final d'explicació

Per a cada string d'entrada, haureu de donar com a sortida el resultat d'aplicar sobre aquesta entrada el següent conjunt de regles de reemplaçament amb una derivació per l'esquerra, és a dir, aplicant a cada pas una regla el més a l'esquerra possible:

- $n \rightarrow E$
- $v \rightarrow E$ no es pot aplicar si ve seguida de =
- $E * E \rightarrow E$
- $E + E \rightarrow E$ no es pot aplicar si ve seguida de *
- $E < E \rightarrow E$ no es pot aplicar si ve seguida de * o +
- $v = E ; \rightarrow I$
- $(E) \rightarrow E$ no es pot aplicar si ve precedida de i o w
- $i(E)I \rightarrow I$ no es pot aplicar si ve seguida de e
- $i(E)IeI \rightarrow I$

- $w(E)I \rightarrow I$
- $LI \rightarrow L$
- $I \rightarrow L$ no es pot aplicar si ve precedida de $)$ o L
- $\{L\} \rightarrow I$
- $L \rightarrow P$ només es pot aplicar si aquest L és l'únic caràcter que queda, és a dir, si tot l'string inicial ha estat reduït a exactament L .

Aquestes regles tenen una pinta una mica críptica. A continuació donem una petita explicació del seu significat i el sentit de l'exercici. Remarquem, però, que aquesta explicació no és imprescindible per a poder resoldre l'exercici. La podeu obviar si voleu.

Explicació (la podeu ometre)

Aquestes regles son una manera simplificada de descriure com reduir una seqüència de tokens d'un cert llenguatge de programació. Les lletres majúscules tenen el següent significat intuïtiu:

- E: expression
- I: instruction
- L: list of instructions
- P: program

Per exemple, la regla $i(E)IeI \rightarrow I$ es pot interpretar com:

if (expression) instruction else instruction \rightarrow instruction

Final d'explicació

Nota: el sistema de regles anterior acaba, però no és convergent. Per exemple, l'string d'entrada $v=n;v=n;$ té dues formes normals LL i P , com es mostra amb les següents dues derivacions:

- $v=n;v=\underline{n} \Rightarrow v=n;v=\underline{E} \Rightarrow v=n;\underline{I} \Rightarrow v=\underline{n};L \Rightarrow v=\underline{E};L \Rightarrow \underline{IL} \Rightarrow LL$
- $v=\underline{n};v=n; \Rightarrow v=\underline{E};v=n; \Rightarrow \underline{I}v=n; \Rightarrow Lv=\underline{n} \Rightarrow Lv=\underline{E} \Rightarrow \underline{LI} \Rightarrow \underline{L} \Rightarrow P$

Tot i així, hem dit que volem el resultat d'anar aplicant les regles el més a l'esquerra possible. Per tant, només la P , que és el resultat de la segona de les derivacions anteriors, es consideraria com a sortida vàlida del programa amb entrada $v=n;v=n;$.

Observació: Podeu seguir l'enfoc que considereu oportú, i podeu utilitzar qualsevol de les classes presentades al curs (**string**, **vector**, **stack**, **queue**, **list**, **map**, **set**) de la manera que considereu oportuna. Noteu, però, que enfocaments diferents poden donar lloc a solucions més o menys eficients, i que superin només els jocs de proves públics o tots els jocs de proves, de manera que la nota acabarà depenent d'això.

Recomanació: Aquest exercici pot ser bastant tediós de resoldre si no s'agafa un enfocament convenient. Considerem que és recomanable utilitzar un `vector` v com si fos una pila, enlloc de fer servir un `stack` directament. El motiu és que el tipus `stack` només permet comprovar el valor del cim de la pila, però no del segon valor des del cim, del tercer valor des del cim, etc, sense haver de desapilar elements. L'enfocament que us recomanem usant v és el següent. Teniu un bucle general que recorre els caràcters de l'string d'entrada. A cada pas del bucle general, afegiu un nou caràcter a v amb `push_back`. Dins del bucle general, teniu un bucle intern que intenta aplicar regles al final de v mentre sigui possible.

A cada pas del bucle intern, verifiqueu si el final de v coincideix amb una part esquerra de regla, i satisfà les condicions per a poder aplicar aquesta regla. Noteu que les condicions sobre caràcters que precedeixen la part esquerra de regla s'hauran de comprovar sobre el propi v , mentre que les condicions sobre caràcters que segueixen la part esquerra de regla s'hauran de comprovar sobre el que queda per llegir de l'string d'entrada. Quan poguem aplicar alguna regla, modificarem el final de v reemplaçant la part esquerra de regla per la corresponent part dreta.

Entrada

L'entrada conté un nombre arbitrari de casos, un per línia. Cada cas consisteix en un string no buit sobre $\{+, *, =, <, \{, \}, (,), ;, v, n, i, e, w\}$.

Sortida

Per a cada cas, escriviu en una línia el resultat d'aplicar les regles de reemplaçament tant com sigui possible i sempre donant prioritats a aplicar regles el més a l'esquerra possible (o sigui, la forma normal obtinguda amb una derivació per l'esquerra).

Exemple d'entrada 1

```
v=n; i (n) v=n; i ( (n) *n) v=n; i (n) v=n; ev=n; v=n;
i (v) i (v) w (n) w (n) v=n; ev=n; e {v=n; v=n; }
v=n*n*n*n+n+n<v*v; i (n*n*n) {v=n; }
{ }=n; {vnn; v=n; } i (n) v=n; } v= ( (n)n+n*n<n) );
w (n) v=n<n*n; i (n*+) {w (n) v= } ;
i ( (n) ) i (n<n*n+n) v=n; ev=n; ei (n) v=n; ev=n; v=
v=n; i (n) {v=n; *w ( (n) ) } i (n<n+ ( * (n) <n*n<n) ) v=n;
i (v) i (n+n) v=n; ew (n) vnn; v=n; v=n;
w (n) v=n; i (n+v) {v=n; } ei (n+n) w (n) w (n) v=n; e {v=n; }
{ i (n) i (n) v=n; ( {v= (n+n) } ; } v=n;
v=n; v=n; v=n; i (n) v=n; i (n) {v=n; } ev=n; v=n;
i (n* ( (n) ) <n* (n) ) {v=n; } ev=n; v= ( {v=n; {v=n; } iw (n) i (n) v=n; }
i (n<v) i (v) v=n; ew ( { } ) w (n) v=n; v=n;
v=n<i (n<n* (n) *n<v*n) v=n; ev=n; w (n) v= } *n;
{v (n; } v=v; v= ) ; i (n) w (n) v=w; ev=n; v=n;
i ( (n<n=n+n+ (n<n) +n) ) {v=n; } i (n*n) v=n; e=n;
v=n; v=n; i (n) v=n; ev=n; i (v) i (n) {v {n; v=n; } ev=n;
v; v; i (v) v= ( ( (v< (n<n) +n) *vwn+n+v) ) ;
i (v) {v=n; w (n) i (n) v=n; } ei (n+n) {v=n; } v=n;
v=n; v=n; w ( (n+n) <n+n+< ) i (n*n+ (n) ) v=n;
i (n<v) n; v=n; v=n; w ( (n) ) i ( (n) ) vnn; v=n; v= (n*n*n) ;
v=n; v=n; i (n) v=n; ev=n; w (v) v=v; i (n+n*n) {v=n; {v=n; } } w (n) v=v;
{v=n; v=n+i (n {n) i (n) (=n; ev=n; ei (n) v=n; ev=n; } w (n) w ( (n) ) w (v) v=n;
v=n; w (n) v=n; v=n; v=*; w (n) {v=w; v=n { } v= (n) +v;
*=n; i (n) v=n; ev=n; i (n) v=n; ev=n; vvn; i (n) w (n) i (n) v=n;
w (n+n<n*n) v=n; i (n+n<n) v=n+v<n+n<n;
i ( ) v=n; e (n) w (n) v=n; v=n;
v=n+ (v) +n+n+n+n<n) ; {v=n; } vv=n; i (n) v=*; ev=n; }
i (n) w { } w (n<n) v= (n) ; v=n;
{v=n; } v=n; i (v) {v=n; v=+; v=n; i (n) v=n; ev=n; } ev=n;
{v=n; } v=n; {v=n; } i ( (n) +n<n) v=n; =i (n) v=n; ev=n; i (v) {v=n; =
v=n; i (n*n) w ( (n) ) i (n) v=n; ev=n; w (vn) ) {v<n; }
v=n; v=n; {v=n; =v=n; v<n; i (n<n) {v=n; } v=n; v=n;
v=n; v=n; i (n) i (n) i (n) ) =n; ev=n; ei ( { } ) v=n;
v=n; v=n; v=n; v=n; i (n) v=n; i ( (n) ) v=n; ei (n) =n;
w (n*n) w ( ( (n) <n) ) w (n+n) {v=n} v=n; v=n; } {v=n; v=n; {v} n; } i
```

```
v=n; { i (n) w (n) i } (n**)) {v=n; } e {v=n; } }
v=n; i (v) i (n) v=n; ev=n; ev=n; i (n) v=n; e {v=n; v=n; } i (n) v=n; e
v+v; w (v*n*n*n) i (n+n+v=n;
v=n; i (v) i (n) v=n; i (n) v=n; i ( ( ( * ( (n) ) ) ) *v=n; } ei (n) v=n; ev=n (
w (n) w ( ( (n) ) ) v=n; i ( ( ( * ( (n) ) ) ) *v=n; } ei (n) v=n; ev=n (
```


<pre> } (n<n) v=n; w (n) i (n) v=n; i (n) v=n; ei (n) i (n*n+n) v=n; w (n<n) i (n*n*n) i (n) i= (; ev= (n*; ei (n) v=*) i (e) w ((n) * (<nwi (n) v=n; v=n*n; v=n; i (n) {v=n; w (v+n) {v= (v) ; } i (;) {v=n; v=n; } e {v=n}) =n; } w v=n; i ((v) *n) w (n*n+n) v=n; e+v=n; } v=n; {w (v) i </pre>	<p>Exemple de sortida: 2</p> <pre> ; ev=n; v=n; v=n; i (n) v=n; v= (; i ((E=) v= (; i (L) *E) Iew (E) v= (; i (E) eELELi (;) v=Eiel i (w (n*n+n) <n<n<n*n*n) v=v+; v=n; v=n; i (Lw (L) E) *i) L) L v=E; L v; P (n) v=n; v=n; } v=n; w (n<n<n) v=v; LEELELE; L} EL Lv=E} L Li (E) E {E; i (; =LeEEE) ILEEL {LEL Lv=E<i<E; LE=*; w (E) w (E (L Li () E)) {i (E) Iev= (E+w) e} w {E< (w+E) } EEE; L Lv= (; P P L {v=*; } {w (<) E=E; } i (E) i (E) Iei *E) IeLw (i) v=Ee L {=E; LE {E; Li (=) Ie {v=E=v=; ; } L Li (E) EEE; eLi (E) i {E) i (E) < ({ I<LeLi (E) w (E) {; =E; } Lw (E) v= (+) i (E) Ie=ELe= (+) IEiE; P i (E) +EL {w (E<L) v= (=E) } E; Lw (E) v=; } L=ELw (E) i (E) Iev=<; L Lw (E) *=E; Lw (E)) I} Lw (E) i (E) {E {E; L} eLw (E) Li (E) i (E<L {L<=E; v=E {E; } Lw (E) i (E) {v=E () LE; Lw (E+ (E+) {E (E; EeE; L} L i (E) {i (E) i (E) E) E; eL} eL {Li (EELeL} Lw (E) {=E; L (EL=L (=E; L v=*; EwE; L {Li (<*E) I} Lv=; ; i (; < ((EE)) I Li (E) ; E+) EEE; eLev=E+Lw (E) i (E) i (E+ {<E) IeLeL L (ELv=*; LwEE) IL Lw (E) i (E) w (E) v=EEeL) ELeL i (E) Iev=e; i (E) i (E) Ie; =E; eLi (E) ==E; L Li (E) IewEE) Ii (E) {L=eLv= (<) ; {v=EE} i (E) w (E) i=E) *=E; eLeLv=EEE; v= (w) ; i (E) i (E<*<E) IeL L+=E; L+EL {Lii (E) i (E) i (E) Ie {LEeE; } i (E (LeL) =E; L {Li (E<; *E) i (E) v=<; } L Lv=EEL Lv=E*L P i (E) {Ee=; L} e; eE; Li (E) i (E) IeiEEEE) v=E* } ; L EEL {EE<; L} L {w (E) v=E=i (E) i (E) v=E=ev=* } L L) =E; Li (E) w (E< (E)) IL {v=E () L L (EE) {Li (E) v=E<; ; } {v=E=} i (E) {EwE; {EL} eL i (E) i ()) ILw (EeL P P L+=i; v=w; i (E) v=E+eLE; w (E) i (E) i=E) Iw (E) w (E) EEE; Lw (i *E) IL i (E) i (E) i (E) Iev=EEE; L wiE) ILi (E) i (E) Ie *Ew (E) v=*; e {E) E; L} i (Eww (i+wE)) ILE; L (Ee; {Lw (e) I} i (E) i (E) i (E)) ELeLew *E) IL P Li (E<E+i<E) Iv=EeLi (E) } L} L Li (E+ (=) +E) () =E; } w (E+i (E) v=Eeei (*) IeLi (E) {E=E; } ew (E) w (i (E) i) E) IeL) EL L {v=w; } {; =E; } =ELEe=E; E {E<*; {i (E) Ie) ELeL} Lw (E=Lw (E) {i (eE)) I} i<E) IL i ((E+) Ii (E) i ((*) +E) Iw (E) w (E) i (E) +=E; eLv=; ; w (Ew+=E) v=<; i (E) w (E) w (E) w (E+L P Li (EE) =; } L {i=E; } {E=E; L} L; =E; i (E) LeLi (E) =EL L) =E; v=i; Li (({}) i (E) Iei (E) ieE) IeL Li (w) i {E) IEi (E) EwE; (i ()) IeL </pre>
---	---

```

Li (E) i (E) i (E {Li (E) {Li (EEL) eL
L {E} E; L} w (EE {v=E} } L {Li (E) Ie +=E; L} L
Lw =E; v=) ; L+L+Lw ( ( () + ( (e<E) ) ) I
e =E *Lv =E *v =E <=E) +E; L
i (E) wiE Ii =E) IL {L { * =E; L} } L} L
Li *E) IL {E =E; } v =E =w (E) v =E) IL
Li ( (E <w) ) Iei (E) w (E) ==EEeL
P
LEw (E *L {L
Li (E) i (E) i (E) w (E) ; E; } Lw ( {} ILi (E) EEL
+ELw (EE (= (+) +E * ) ; L) =E; L

```

```

Lv =E *LEL
LE {E; LE; E; w (E) i +E) IeLeLi (E) Iev =E =L
w =E; Li (E) w ( ) ) IeL
Lw (*E) +E) ILi (E) i (E < (E+) IeL
L <=E; L
} ELi (E) Iei (E) i (E) i (E) Iei ( ( * ) Iw *i) IL
Lw (E) i (E) i (E) i = ( ; ev = (E * ; ei (E) v = * ; eLv = { ;
i (e) w (E * (<EwLi (E) {L {w (E <<<E) v = + ; L
Li ( ; ) Ie {v =E} ) =E; } w (E) i (E) w (E) v =E + ; <E; L
Li (E) Ie +L} L

```

Observació

Avaluació sobre 10 punts:

- Solució lenta: 5 punts.
- solució ràpida: 10 punts.

Entenem com a solució ràpida una que és correcta, de cost lineal i capaç de superar els jocs de proves públics i privats. Entenem com a solució lenta una que no és ràpida, però és correcta i capaç de superar els jocs de proves públics.

Informació del problema

Autor : PRO2

Generació : 2024-03-08 17:58:23

© *Jutge.org*, 2006–2024.

<https://jutge.org>