Pràctica de PRO2 - Tardor 2016 (provisional) X93782_ca

Aquest és un problema de Jutge que permet fer lliuraments de prova de la pràctica. Tingueu en compte que:

- No és el canal per a fer el lliurament definitiu de la pràctica.
- El lliurament definitiu inclourà més fitxers que els que es demanan aquí.
- El lliuraments fets en aquest problema del Jutge no seran tinguts en compte per a la nota de la pràctica.

Entrada

Una seqüència d'instruccions seguint el format de l'enunciat de la pràctica i del joc de proves public.

Sortida

El seu resultat seguint el format de l'enunciat de la pràctica i del joc de proves public.

Observació

El Jutge prova el vostre lliurament mitjançant 4 jocs de proves:

- public: el joc de proves públic.
- privat1: joc de proves privat que fa èmfasi en l'avaluació d'expressions construits combinant enters, llistes d'enters i operacions primitives.
- privat2: joc de proves privat que fa èmfasi en la definició de variables i funcions senzilles, i en l'avaluació d'expressions que combinen constants, variables, operacions primitives i funcions definides.
- privat3: joc de proves privat que fa èmfasi en la definició de funcions més complexes, i en l'avaluació d'expressions que inclouen crides à funcions definides i operacions primitives.

En un fitxer de nom `practica.tar` heu de lliurar

- Els fitxers .hh i .cc.
- El fitxer Makefile (l'usarem per genera el fitxer executable i provar-lo).

Tingueu en compte les restriccions següents:

- El mòdul que conté la funció `main` s'ha de dir `program.cc`.
- El `Makefile` ha de generar un executable de nom `program.exe`. El Jutge internament executarà la comanda `make program.exe`.

- Recomanem que useu les opcions de compilació del Jutge de PRO2 (vegeu Documentation → Compilers → PRO2 a www.jutge.org). Altrament us arrisqueu a patir dos tipus de problemes: excés de temps durant la compilació (`compilation time exceeded`) o excés de temps durant l'execució.
- No usar l'opció `-D_GLIBCXX_DEBUG` o no usar-la correctament serà fortament penalitzat.

Produïu el fitxer `.tar` amb la comanda

```
tar -cvf practica.tar fitxer1 fitxer2 fitxer3 ...
```

des del directori on es troben els fitxers que heu de lliurar. Poseu aquesta instrucció en el vostre `Makefile` de forma que es pugui generar el .tar executant `make practica.tar`. Amb això reduireu la possibilitat d'error en enviaments successius. El Jutge no accepta `.tar` on els fitxers a lliurar es troben dins de carpetes. Recomanem usar `GNU tar` per reduir el risc que el .tar singui incompatible amb el Jutge.

Exemple d'entrada 1

```
0
-1
666
-90127
()
(3)
(-25)
(23 4)
(-12 -71)
(0  -10   11)
(1 -1 1 -1 1 -1 1)

(+ 10 15)
(+ (+ 1 2) 7)
(+ 10)
(+ 10 (15))
(- (- 5))
(+ x 1)
(cons 3 (2 1))
(define l1 (20 30))
(head (tail (cons 10 l1)))
(head (tail (tail l1)))
(if 1 (+ 0 1) (+ 2 3))
(if 0 (+ 0 1) (+ 2 3))
(if 2 (+ 0 1) (+ 2 3))

(define x 10)
(define z (+ (head (tail (1 2 3 4))) 10))
(+ (head (1 2 3 4)) z)
(if (< z x) (1) ())

(define diff (x y) (+ x (- y)))
(diff 10 20)
(define * (x y) (if (= x 0) 0 (+ y (* (diff x 1) y))))
(* 4 5)
(define * (x y) (if (= x 0) 0
                    (if (< 0 x) (+ y (* (diff x 1) y))
                                (diff (* (+ x 1) y) y)
                    )
                )
)
(* 4 5)
(* -4 5)
(* 4 -5)
(* -4 -5)
(define quadrat (x) (* x x))
(quadrat 5)
(define / (x y) (if (< y x) (+ 1 (/ (diff x y) y))
                            (if (= x y) 1 0)))
(/ 21 4)
(define sum-first (n) (if (< 0 n) (+ n (sum-first (diff n 1
(sum-first 10)
(define mitjana (n) (/ (sum-first n) n))
(mitjana 10)

(define <= (a b) (or (< a b) (= a b)))
(define >= (a b) (or (< b a) (= a b)))
(define == (a b) (= a b))
(define > (a b) (< b a))
(and (<= x x) (>= x z))
(not (== x z))
(> z x)
(<=> z x)

((+ x z) -10 (head ((diff z x))))
(1 (head (cons -3 ())) (- (/ z x)) (if (< x 5) () 99))
(1 (head (cons -3 ())) (- (/ z x)) (if (< x 5) 99 ()))

(define abc (+ x z))
(define xyz (* abc 10))
(* abc xyz)
(define abc 123)
(* abc xyz)
(define fun (y) (= y (* (/ y x) x)))
(fun 10)
****
```

Exemple de sortida 1

```
0
-1
666
-90127
()
(3)
(-25)
(23 4)
(-12 -71)
(0 -10 11)
(1 -1 1 -1 1 -1 1)
25
10
indefinit
indefinit
5
indefinit
(3 2 1)
l1 (20 30)
20
indefinit
1
5
indefinit
x 10
z 12
13
()
diff #2
-10
* #2
20
* #2
20
-20
-20
20
quadrat #1
25
/ #2
```

```
5
sum-first #1
55
mitjana #1
5
<= #2
>= #2
== #2
> #2
0
1
1
indefinit
(22 -10 2)
(1 -3 -1 99)
indefinit
abc 22
xyz 220
4840
abc 123
27060
fun #1
indefinit
Variables:
abc 123
l1 (20 30)
x 10
xyz 220
z 12
Operacions:
* #2
/ #2
<= #2
== #2
> #2
>= #2
diff #2
fun #1
mitjana #1
quadrat #1
sum-first #1
```

Informació del problema

Autoria: Professors de PRO2

Generació: 2026-01-25T21:34:06.689Z