

---

## Pràctica de PRO2 - Tardor 2016 (lliurament definitiu) X74817\_ca

---

**En negreta s'indiquen les novetats respecte el problema del Jutge per fer el lliurament provisional de la pràctica.**

Aquest és el problema de Jutge que permet fer el **lliurament definitiu de la pràctica**.

Avisos importants:

- **En l'avaluació de la pràctica només tindrem en compte els lliuraments realitzats pel membre de l'equip que és el responsable de fer els lliuraments en aquest problema del Jutge (acordat amb el vostre professor de laboratori).**
- **Els enviaments fets per altres membre de l'equip o en el problema del Jutge per al lliurament provisional no seran tinguts en compte.**
- **Només avaluarem el darrer enviament realitzat pel responsable de l'equip dins el termini establert.**
- No superar cap joc de proves suposa una nota de zero de tota la pràctica.

### Entrada

Una seqüència d'instruccions seguint el format de l'enunciat de la pràctica.

### Sortida

El seu resultat seguint el format de l'enunciat de la pràctica.

### 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 construïts 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 a funcions definides i operacions primitives.

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

- Els fitxers `.hh` i `.cc` **de les classes que heu implementat (amb els comentaris de Doxygen escaients).**
- El fitxer `Makefile` (l'usarem per generar el fitxer executable i provar-lo).

- Un joc de proves per cada membre de l'equip. Cada joc de proves consisteix en una entrada i una sortida esperada. L'usuari `usuari.est.fib.upc.edu` ha de produir dos fitxers de nom:

```
jp_usuari_entrada.txt
jp_usuari_sortida.txt (sortida esperada)
```

- Un document de nom *equip.pdf* on s'especifiqui la repartició de la feina entre els membres de l'equip, per exemple, quins mòduls, mètodes i jocs de proves ha fet cadascú. Aquest document ha de tenir una extensió màxima de dues pàgines (amb tipus de lletra d'11 punts). A l'inici d'aquest document s'han d'indicar els membres de l'equip (nom, cognoms i usuari UPC de cadascun). El primer membre esmentat ha de ser el responsable dels lliuraments.
- Un fitxer `html.zip`, obtingut zipejant la carpeta `html` del doxygen generat a partir dels `.cc` i els `.hh` dels mòduls. La documentació en Doxygen no sols ha de cobrir la part pública de les classes (com al lliurament de l'especificació) sinó també la part privada (atributs i mètodes privats).

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 **Flags1** de compilació del Jutge de PRO2 (vegeu [Documentation](#) → [Compilers](#) → [PRO2](#) a [www.jutge.org](http://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.
- El `.tar` que lliureu al Jutge no pot superar 4Mb.

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 reduïreu 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` sigui incompatible amb el Jutge.

### Exemple d'entrada

```
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)
```

<pre> (+ 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 (&lt; 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 (&lt; 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 (&lt; y x) (+ 1 (/ (diff x 1) y))                     (if (= x y) 1                         (/ x y)                     ) ) (/ 21 4) (define sum-first (n) (if (&lt; 0 n) (+ n (sum-first (diff n 1))) 0)) (sum-first 10) (define mitjana (n) (/ (sum-first n) n)) (mitjana 10)  (define &lt;= (a b) (or (&lt; a b) (= a b))) (define &gt;= (a b) (or (&lt; b a) (= a b))) (define == (a b) (= a b)) (define &gt; (a b) (&lt; b a)) (define (&lt;= x x) (&gt;= x z)) (not (== x z)) (&gt; z x) (&lt;=&gt; z x)  ((+ x z) -10 (head ((diff z x)))) (1 (head (cons -3 ())) (- (/ z x)) (if (&lt; z 5) () 99)) (1 (head (cons -3 ())) (- (/ z x)) (if (&lt; 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) **** </pre>	<h3 style="margin: 0;">Exemple de sortida</h3> <pre> 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) 11 (20 30) 20 indefinit 1 5 indefinit x 10 z 12 13 () diff #2 -10 x #2 20) * #2 20 -20 -20 20 quadrat #1 25 / #2 5 sum-first #1 55 mitjana #1 5 &lt;= #2 &gt;= #2 == #2 x #2 x 5) 99 ())) 1 1 indefinit (22 -10 2) (1 -3 -1 99) indefinit abc 22 xyz 220 4840 </pre>
---	---

```
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

Autor : Professors de PRO2  
Generació : 2016-12-16 21:57:15

© *Jutge.org*, 2006–2016.  
<http://www.jutge.org>