

Trabajo Práctico N° 3

Intérprete de pseudo C en LISP

"Máquina virtual de *pseudo C* en LISP"

Fecha de entrega:

Escribir una función en LISP que llamaremos RUN, que interpreta un programa escrito en un lenguaje similar a C, lenguaje Pseudo C. La función RUN recibe dos parámetros, uno es una lista que representa el programa a interpretar y el otro parámetro es una lista de datos que contiene números a consumir por las lecturas que contenga el programa. La función debe devolver una lista con todos los números que se imprimen en el programa. En caso de haber un error en cuanto a variables no declaradas, debe devolver el mensaje de error.

La sintaxis a utilizar para escribir un programa en pseudo C es:

- un programa es una lista que contiene sublistas
- las declaraciones de variables enteras (solo hay variables enteras) están contenidas en sublistas
- las instrucciones son sublistas que están contenidas dentro de la sublista main
- se podrá utilizar las funciones “printf” y “scanf” (solo con variables enteras)
- las estructuras de control a reconocer por el intérprete son if-then-else y while
- el código a interpretar no será estrictamente C sino que será una lista de listas con las instrucciones C (código intermedio entre C y LISP)

Ejemplo: código en C

```
int x;
int z, A = 10;
void main() {
  z = A + 1;
  printf("%d", A);
  scanf("%d", &x);
  if (A < X) {
    z += A;
  } else {
    z = 1;
  }
  while(x < 10) {
    printf("%d", x);
    x++;
  }
}
```

Código en pseudo C, que recibirá el intérprete

```
(
  (int x)
  (int z A = 10)
  (main (
    (z = A + 1)
    (printf A)
    (scanf x)
    (if (a < X) (
```

```

(z += A)
) else (
(z = 1)
))
(while (x < 10) (
(printf x)
(x++)
))
))
)

```

Ejemplo de la invocación de la función RUN con un programa que calcula el factorial de un número leído.

```

(RUN '( (int n fact = 1)
      (main (
              (scanf n)
              (if (n < 0 )
                  ( (printf "no existe fact de nro negativo" ))
                  else
                  (
                    (while (n > 1)
                      ( (fact = fact * n)
                        (n -- )
                      )
                    )
                    ; cierra while
                  (printf fact )
                  ; cierra else
                )
                ; cierra if
              )
              ; cierra main
            )
            '(5) ; sublista con los valores a leer del programa
          )

```

La función RUN devuelve una lista que contiene el factorial de 5 o sea (120)

Otros ejemplos de llamadas a la función RUN

```

> (RUN '( (int a = 2 b = 3)
          (main (
                  (printf a)
                )
              )
          )
          () )
> -----> (2)

```

```

> (RUN '( (int z = 2)
      (main (
              (printf b)
            )
          )
        ) () )
> -----> (ERROR_VARIABLE_NO_DECLARADA)

> (RUN '( (int a = 6)
      (main (
              (if (a == 2)
                  ( (printf (a + 1))
                    )
              )
            )
          )
        ) () )
> -----> NIL
> (RUN '( (int a = 2)
      (main (
              (if (a == 2)
                  ( (printf (a + 1))
                    )
              )
            )
          )
        ) () )
> -----> (3)

> (RUN '( (int a = 2 b)
      (main (
              (scanf b)
              (a = b + 3)
              (printf a)
            )
          )
        ) '(5) )
> -----> (8)

> (RUN '( (int a = 2 b)
      (main (
              (a = (a + 1) * 4)
              (b -= 5)
              (a += 3)
              (printf a)
              (scanf a)
              (printf a)
              (printf b)
            )
          )
        ) '(6) )
> -----> (15 6 -5)

```

```

> (RUN '( (int x y p = 10)
          (int r)
          (main ( (x = p + 10)
                  (p ++))
                (++ x)
                (x *= p - 4)
                (if (x < p)
                    ( (printf x + p)
                      (scanf y)
                    )
                  else
                    ( (x = x * 6)
                      (printf p * p)
                    )
                )
                )
          (while (x > p * 10)
            (
              (printf x + p)
              (scanf y)
              (printf y)
              (x -= y)
            )
          )
        )
      ) '(700 100) )
> -----> (121 893 700 193 100)

```