Compiladors - Practica 3

Practica de compiladors. Desenvolupada sota l'entorn proporcionat per docker. Com a funcionalitat extres s'ha afegit:

- switch instruction (tambe en c3a)
- array list
- continue instruction (tambe en c3a)
- break instruction (tambe en c3a)

S'ha afegit un script python que compara la sortida amb la sortida esperada, destacant les diferencies.

Requeriments - gcc - flex - bison - make - python3

Docker

```
$ docker build . -t compiladors
$ docker run -v $(pwd):/prac -it compiladors /bin/sh
```

Build

```
$ make
```

Clean

```
$ make clean
```

Run

```
$ ./prac3 INPUT_FILE OUTPUT_FILE
```

Run examples

```
$ make test
```

```
FILE=program_statement_if_8 && make clean && make && ./prac3 ./input/$FILE.txt
./output/$FILE.txt && python3 diff.py --from ./output/$FILE.txt --to
./expected/$FILE.txt
```

Examples

All examples are defined in `inputs` folder.

Documentació

Calc mode

File must begin with calc mode: - `calc on`: begins in calculator mode. - `calc off`: begins in program mode.

Types

int `a := 2`

float `a := 2.0`

string `a := "test"`

array This is a basic implementation of an array list.

```
a := [1, 2.0, 3, 4.0]
a[0]
a[1]
```

Operations

- `+` Add
- `-` Substract
- `*` Multiply
- `/` Divide
- `**` Power
- `mod` Module

Int / float operations

Any operation with floats returns the result as a float.

`2 + 4 = 6`

`2.0 + 4.0 = 6.0`

`2 + 4.0 = 6.0`

String add operation

`"test" + 1234 = "test1234"`

`"test" + "-" + "test" = "test-test"`

Comparators

- > Greather than
- < Less than
- >= Greather than or equals to
- <= Less than or equals to
- = Equals
- <> Different Is it possible to compare any type.

Functions

- `sqrt()`
- `log()`

Program statements

for

```
calc off

b := 2

for (a in b..4) do
  5+5
done
```

if

```
calc off

a := 5

if (a > 5) then
  x:= 2
elsif (a < 5) then
  x:= 3
elsif (a = 5) then
  x:= 5
fi
```

repeat

```
calc off

a := 5

repeat
  5 + 5
  a := a +1
until (a < 7)
```

switch

```
calc off

a := 3

switch (a) do
  case a > 4 then
    4 + 4
    break
  default then
    5+5
    break
done
```

while

```
calc off

a := 5

while(a < 10) do
  5+5
  a := a +1
done
```