

# Definición de subprogramas

$\langle par\_def \rangle \rightarrow type \langle stars \rangle nombre \langle indf \rangle$

$\langle pars\_ant \rangle \rightarrow \varepsilon \mid \langle pars\_ant \rangle \langle par\_def \rangle,$

$\langle pars\_cp \rangle \rightarrow \varepsilon \mid \langle pars\_ant \rangle \langle par\_def \rangle$

$fvnombre \rightarrow V_{nombre} \langle tpars \rangle \mid N_{nombre} \langle tpars \rangle$

$fvnombre_1(\langle pars\_ant_1 \rangle \ type_1 \langle stars_1 \rangle nombre_1[(\langle vindf \rangle)] \langle indf \rangle)\{$



$fvnombre_1(\langle pars\_ant_1 \rangle \ type_1 \langle stars_1 \rangle * nombre_1)\{$

## Ejemplo

$Y_{:foo} \$F \$F(float \text{ bar}, float \text{ baz}[]) \{$



$Y_{:foo} \$F \$F(float \text{ bar}, float * \text{ baz}) \{$

$fvnombre_1(\langle pars\_ant_1 \rangle \langle par\_def_1 \rangle, \langle par\_def_2 \rangle) \{$



$fvnombre_1 \$ \langle par\_def_2 \rangle (\langle pars\_ant_1 \rangle \langle par\_def_1 \rangle) \{$

## Ejemplo

$Y_{:foo} \$F \$F(float \text{ bar}, float * \text{ baz}) \{$



$Y_{:foo} \$F \$F \$float * \text{ baz}(float \text{ bar}) \{$

$fvnombre_1(\langle pars\_ant_1 \rangle) \{ \rightarrow fvnombre_1 \$ \langle pars\_ant_1 \rangle () \{$

## Ejemplo

$Y_{:foo} \$F(float \text{ bar}) \{$



$Y_{:foo} \$F \$float \text{ bar}() \{$

`fvnombre1 $\$Y\$type_1 \langle stars_1 \rangle * nombre_1(\langle pars\_cp_1 \rangle) \{ \rightarrow$`

`fvnombre1( $\langle pars\_cp_1 \rangle$ ) {  
type1  $\langle stars_1 \rangle * nombre_1$ ;`

Ejemplo

`Y:foo $\$Y\$float * bar() \{$`

$\Rightarrow$

`Y:foo $\$Y() \{$`   
float \*bar;

`fvnombre1 $\$Y\$unsigned int nombre_1(\langle pars\_cp_1 \rangle) \{ \rightarrow$`

`fvnombre1( $\langle pars\_cp_1 \rangle$ ) {  
unsigned int nombre1;`

`fvnombre1 $\$Y\$float nombre_1(\langle pars\_cp_1 \rangle) \{ \rightarrow$`

`fvnombre1( $\langle pars\_cp_1 \rangle$ ) {  
float nombre1;`

Ejemplo

```
float power(float base, float n){  
    float i, p;  
    p = 1;  
    for(i = 1; i <= n; ++i){  
        p = p * base;  
    }  
    return p;  
}
```

$\Rightarrow$  ...  $\Rightarrow$

`F:power $\$F\$float n(float base) \{$`   
float i, p;  
p = 1;  
for(i = 1; i <= n; ++i){  
 p = p \* base;  
}  
return p;  
}

$\Rightarrow$

`F:power $\$F(float base) \{$`   
float n;  
float i, p;  
p = 1;  
for(i = 1; i <= n; ++i){  
 p = p \* base;  
}  
return p;  
}

$\Rightarrow$

```

F:power$F$float base(){
    float n;
    float i, p;
    p = 1;
    for(i = 1; i <= n; ++i){
        p = p * base;
    }
    return p;
}

```

⇒

```

F:power(){
    float base;
    float n;
    float i, p;
    p = 1;
    for(i = 1; i <= n; ++i){
        p = p * base;
    }
    return p;
}

```

⚠ *Nota sobre C:* la posibilidad de usar variables globales no está recogida en las macros.

```

type (stars)*V:nombre() { → Y:nombre() {

```

```

unsigned int V:nombre() { → Y:nombre() {

```

```

float V:nombre() { → F:nombre() {

```

```

V:nombre() { → texto2
    texto1    $V:nombre() {
}              texto1
texto2        }
EOF           EOF

```

## Ejemplo

```

F:power(){
    float base;
    float n;
    float i, p;
    p = 1;
    for(i = 1; i <= n; ++i){
        p = p * base;
    }
    return p;
}
...

```

⇒

```

...
$F:power(){
    float base;
    float n;
    float i, p;
    p = 1;
    for(i = 1; i <= n; ++i){
        p = p * base;
    }
    return p;
}

```