

E-fólio B: máximo 4 valores

Considere a linguagem PFNTleC (Por Favor Não Tente Isto em Casa) com as características explicadas no e-fólio A. No e-fólio B, deverá fazer a geração de código intermédio (para TAC) e otimização de código. Pode, também, melhorar o e-fólio A (ou fazer de início, caso não tenha entregue) tendo em conta os comentários, sendo a nota alterada em conformidade. Pegando no exemplo de código dado no e-fólio A:

// NOTA IMPORTANTE: pode haver "erros de copy-paste" pelo que convem esclarecer qualquer duvida sobre o código gerado ou otimizado deste exemplo

```
// number(x)
```

```
// number(var1,weekDays)
```

```
// vector(y,5)
```

```
// vector(z,9/2)
```

```
// nenhum código TAC até agora
```

```
//value(weekDays, 7)
```

```
_t0 = 7
```

```
weekDays = _t0
```

```
//value(x,weekDays*10)
```

```
_t1 = weekDays * 10
```

```
x = _t1
```

```
//value(var1,x*2+5)
```

```
_t2 = x * 2
```

```
_t3 = _t2 + 5
```

```
var1 = _t3
```

```
// ivalue(y,0,1)
```

```
_t4 = 0
```

```
_t5 = _t4 * 4
```

```
_t6 = 1
```

```
y[_t5] = _t6
```

```
// ivalue(z,0,y(0))
```

```

_t7 = 0
_t8 = 4 * _t7
_t9 = 0
_t10 = 4 * _t9
_t11 = y[_t10]
z[_t8] = _t11

// (number(j,k),value(j,0),value(k,1))
_t12 = 0
j = _t12
_t13 = 1
k = _t13

//number(i)
// for(i,1,size(y)-1, ivalue(y,i,i*y(i-1)))
_t14 = 1
i = _t14
// size(y) = 5
L0:
ifZ i<= 5-1 goto L1 // gralha corrigida
_t15 = i
_t16 = _t15 * 4
_t17 = i - 1
_t18 = _t17 * 4
_t19 = i * y[_t18]
y[_t16] = _t19
i = i + 1
goto L0
L1:

// value(i,1)

```

```
_t20 = 1
i = _t20

// size(z) = 4
// while(i<size(z),(ivalue(z,i,i), value(i,i+1))
L2:
ifZ i < 4 goto L3
_t21 = i
_t22 = _t21 * 4
_t23 = i
z[_t22] = _t23
_t24 = i + 1
i = _t24
goto L2
L3:

// if(j<k,value(j,k),value(k,j))
ifZ j < k goto L4
_t25 = k
j = _t25
goto L5
L4:
_t26 = j
k = _t26
L5:

// if(x<100 and x>0, value(x,x+100))
_t27 = x < 100
_t28 = x > 0
ifZ _t27 && _t28 goto L6
_t29 = x + 100
```

```
x = _t29
```

```
L6:
```

```
// if(j == k,
```

```
// value (x , 100),
```

```
// value (x , 100 - (k + j))
```

```
// )
```

```
ifZ j == k goto L7
```

```
_t30 = 100
```

```
x = _t30
```

```
goto L8
```

```
L7:
```

```
_t31 = k + j
```

```
_t32 = 100 - _t31
```

```
x = _t32
```

```
L8:
```

```
// OTIMIZACAO DE CODIGO
```

```
// O codigo gerado apos otimizacao
```

```
weekDays = 7
```

```
x = 70
```

```
var1 = 145
```

```
y[0] = 1
```

```
z[0] = 1
```

```
j = 0
```

```
k = 1
```

```
i = 1
```

```
L0:
```

```
ifZ i >= 4 goto L1
```

```
_t0 = i * 4
```

```
_t1 = i - 1
_t2 = _t1 * 4
y[_t0] = i * y[_t2]
i = i + 1
goto L0
L1:
i = 1
L2:
ifZ i < 4 goto L3
_t4 = i * 4
z[_t4] = i
i = i + 1
goto L2
L3:
j = 1
x = 170
x = 100
```

// deve entregar um ficheiro ZIP com:

// ficheiros flex e bison

// outros ficheiros que possa ter usado, como ficheiros em C ou de teste

// ficheiro README.TXT com indicacoes para compilacao e execucao do programa

// relatorio descrevendo o vosso trabalho, problemas surgidos, opcoes tomadas, etc.

// BOM TRABALHO!