

fsubs (expr) x = float

type 1

YFSUBS:BSR XIFAC

→ forme factorisée

MOVE #<1, -(SP) arrêt

~~GART0: BSR DECCRV~~
~~BSR GART6~~
~~BSR WVAR @118~~

MOVE TCMPLX, D2
BEQ GART0
SUBQ #4, SP
MOVE D2, -(SP)
BSR LB16
BSR XICFLA

si complexe : i = 1w i

→ non

proce d2
converti en flottant

GART0: BSR DECCRV
~~BSR GART6~~
BSR WVAR @118

⊗

→ di
décode x (littéral) → d1

~~MOVE D1, -(SP)~~
~~BSR DECCREE~~
~~BSR WFLOAT~~
~~BRA GART0~~

MOVE.L SP, A0
GART2: MOVE (A0)+, D0
BMI GART4
ADDQ #4, A0
CMP D0, D1
BNE GART2
BRA ERNR

littéral répété

NOM Répété

GART4: SUBQ #4, SP
MOVE D1, -(SP)
BSR DECCREE
BSR WFLOAT
BRA GART0

⊕ on encode
décode un flottant

```

GAR76: MOVE TVARN, D4
      MOVE.L SP, A1

```

met les adresses des flottants

```

GAR78: BSR LB95C
      TST (A1)+
      BMI GAR80
      MOVE.L A0, (A1)+
      SUBQ #1, D4
      BRA GAR78

```

```

GAR80: MOVE.L SP, A1
      MOVE D4, -(SP)
      ADDQ #6, A6
      BSR XFSTV
      BSR LC12
      MOVE (SP)+, D0
      BSR WASGN2

```

MOVE.L AS, -(SP) ⊗
 prints liste des substitutions
 a0 : expr_f

← MOVE.L (SP)+, AS ⊗

```

GAR82: TST (SP)+
      BMI GAR84
      ADDQ #4, SP
      BSR POPN
      BRA GAR82

```

```

GAR84: RTS

```