

Entree definit

P1: MOVE.L 4(SP), BASEPAGE

```

CLR.L -(SP)
MOVE #20, -(SP)
TRAP #1

```

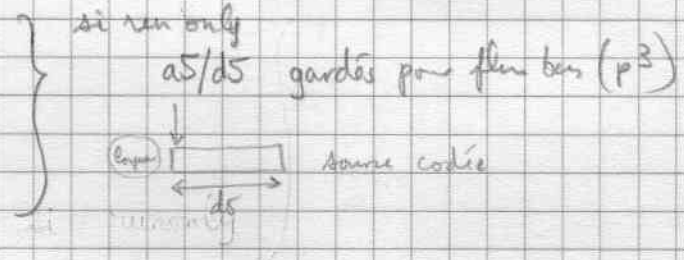
} mode super

net A0 ← 1000 octets apres la demande

```

LEA FIN, A5
LEA 1000(A5), A0
IF FLAGRY:
MOVE.L (A5)+, D5
ADDQ.L #1, D5
BCLR #0, D5
ADD.L D5, A0

```



ENDIF

```

MOVE.L A0, USP
LEA 1000(A0), SP
LEA TRTOP, A6

```

} piles

```

MOVE.L SP, A0
MOVE.L A0, (A6) TRTOP
MOVE.L BASEPAGE, D0
SUB.L D0, A0
MOVE.L A0, -(SP)
MOVE.L D0, -(SP)
CLR -(SP)
MOVE #$4A, -(SP)
TRAP #1

```

MALLOC

(répété)

Mshrink

if flag30

```

MOVEQ #1, D1
CMP # $100, (A6)
BCC VA
MOVEQ #0, D1

```

RAMTT = d1 = $\begin{cases} 0 & \text{dans ram ST} \\ 1 & \text{dans ram TT} \end{cases}$

(utilisé par XALLOC(4) 28d)

```

VA: MOVE D1, RAMTT
MOVE D1, -(SP)

```

```
MOVEQ # $44, D3
```

else

```
MOVEQ # $48, D3
```

endif

```
MOVEQ #-1, D0
```

```
MOVE.L D0, -(SP)
```

```
MOVE D3, -(SP)
```

```
TRAP #1
```

```

SUB.L # $4000, D0
ADD.L D0, (A6) ; TRTOP
MOVE.L (A6), TRTOP1

```

Malloc(-1) ou Mxalloc(-1, RAMTT)
venir ST
venir TT

mémoire disponible

```

if flag30 ← MOVE D1, -(SP)
SUB.L # $160000, D0
endif
SUB.L # $4000, D0

```

~~if flag30~~

```
MOVE D1, -(SP)
```

~~endif~~

```
MOVE.L D0, -(SP)
```

```
MOVE D3, -(SP)
```

```
TRAP #1
```

```
MOVE.L D0, ZALLOC
```

```
CLR.B AESOK
```

Malloc(D0) ou Mxalloc(D0, RAMTT)
venir TT

début de la zone allouée

```

:OR #$700,SR
MOVE.L (A6),A0

```

IF FLAGRY

```

ADD.L D5,A5
PD88:MOVE.L -(A5),-(A0)
SUBQ.L #2,D5
BNE PD88
MOVE.L -(A5),-(A0)

```

si rien only: déplace la source

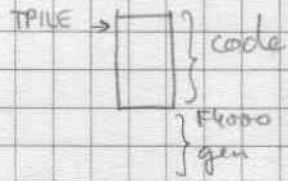
ENDIF

```

MOVE.L A0,D3
MOVE.L A0,SP
LEA MOVE.L PILEUS,A0
MOVE.L A0,USP

```

valeur de la pile pour le moment



```

AND #$23FF,SR
BSR.M VDI100
MOVE.L SP,D3

```

dans VLIMIT: met TPILE TLIMIT TDMAX

```

LEA PILE+1000.L,A6
JSR MASS
JSR IMOOREO

```

initialise le générateur de nombre aléatoires

P2