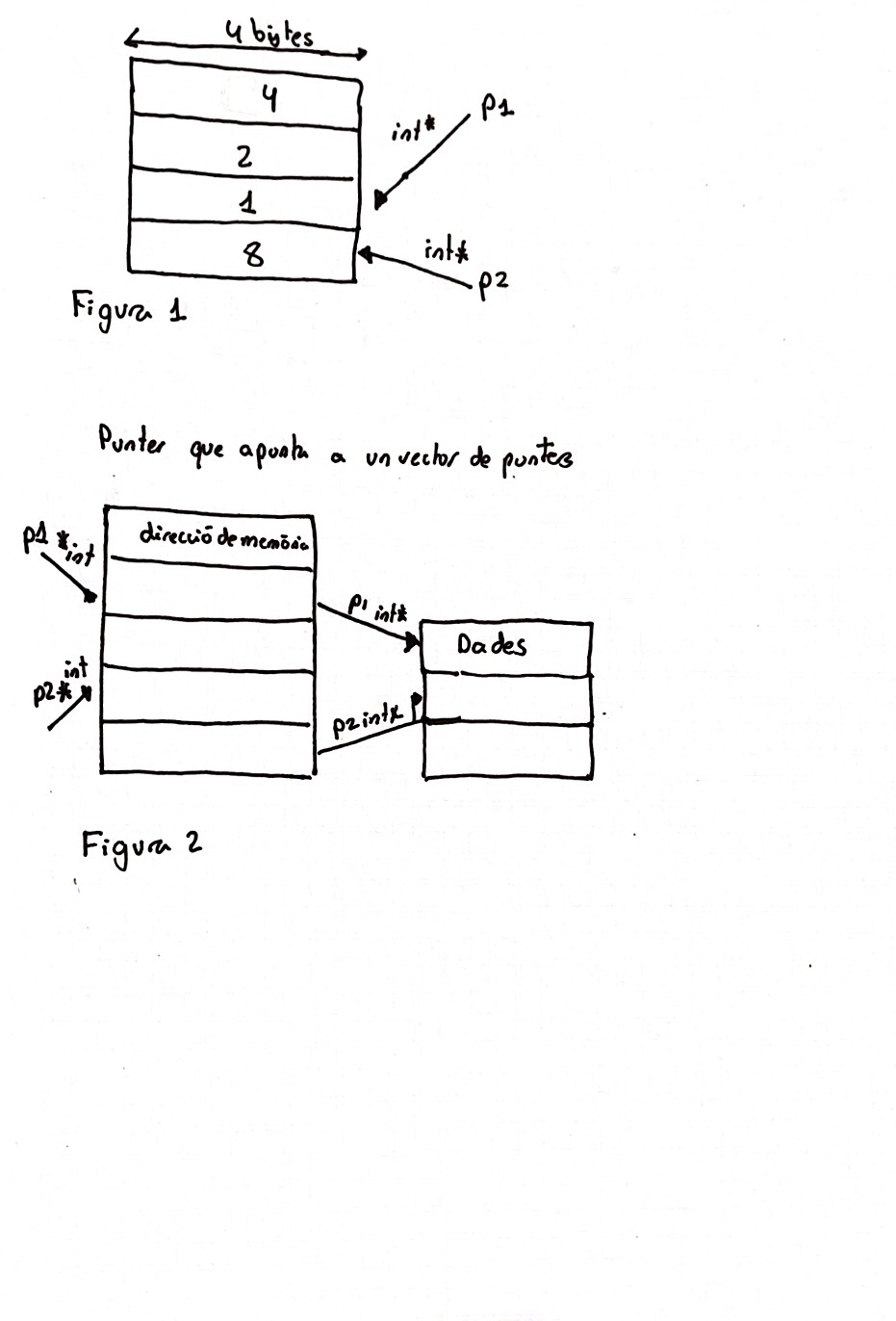
PART A)



1. A la figura 1 podem dir que s'accedeix amb punters (el seu contingut son posicions de memòria), perquè la memòria és un vector de dades dinàmica, i en el seu contingut estan les dades, per això, per poder accedir, només fa falta saber on la posició on està guardat el nostre sencer.

2. A la figura dos, aparentment sembla igual, però canvia el concepte. Al primer pas fem un punter a un vector de punters, és a dir, una posició de memòria que fa referència a una altra posició de memòria on la segona apunta a la posició on tenim guardades les dades. Per això, per accedir a la informació s'accedeix amb doble punter. En resum, en C una matriu dinàmica es declara com un punter de punters, però per entendre-ho millor, és com un vector dinàmic que conté altres vectors dinàmics.

**VALGRIND**

**A)** Si no fem fclose i executem el programa amb –leak-check=full ens apareixen els següents missatges: Ens falten fer un free i ens hem deixat 283,470 bytes a memòria sense alliberar.

HEAP SUMMARY:

in use at exit: 552 bytes in 1 blocks

total heap usage: 10,004 allocs, 10,003 frees, 283,470 bytes allocated

LEAK SUMMARY:

definitely lost: 0 bytes in 0 blocks

indirectly lost: 0 bytes in 0 blocks

possibly lost: 0 bytes in 0 blocks

still reachable: 552 bytes in 1 blocks

suppressed: 0 bytes in 0 blocks

Per saber on s’ha reservat memoria hem provocat errors a fclose i hem executat valgrind amb les comandes extres: --leak-check=full --show-leak-kinds=all.

bytes in 1 blocks are still reachable in loss record 1 of 1

at 0x4C2BBAF: malloc (vg\_replace\_malloc.c:299)

by 0x4E9F55C: \_\_fopen\_internal (iofopen.c:69)

by 0x108917: main (ordStrLen.c:23)

Aquí podem veure que a la linea 23 del main hem reservat memoria que després no alliberem.

**B)** Teniem un error a strcpy (vector[i], aun). Ens passavem sempre d’un byte:

Invalid write of size 1

at 0x4C2EEA7: strcpy (vg\_replace\_strmem.c:506)

by 0x108A76: main (ordStrCon.c:25)

Address 0x51ebbbf is 0 bytes after a block of size 15 alloc'd

at 0x4C2BBAF: malloc (vg\_replace\_malloc.c:299)

by 0x108A4D: main (ordStrCon.c:24)

El motiu era que reservavem memoria fent:

vector[i] = (char \*) malloc ((strlen (aux)) \* sizeof (char));

Però strlen no tenia en compote el caràcter ‘\0’ de aux. Solució: afegir-lo manualment:

vector[i] = (char \*) malloc ((strlen (aux)+1) \* sizeof (char));