

Cálculo científico y técnico con
HP49g/49g+/48gII/50g
Módulo 2: **Recursos avanzados**
Tema 2.5 **Programación User-RPL I**

Francisco Palacios
Escuela Politécnica Superior de Ingeniería de Manresa
Universidad Politécnica de Catalunya
Dep. Matemática Aplicada III

Abril 2008, versión 1.2

Contenido

1. Introducción
2. Variables globales
3. Variables locales
4. Presentación de resultados
5. Ejecución condicionada
6. Formularios de entrada de datos
7. Gestión de la memoria

Índice General

1	Introducción	1
1.1	Programas en User-RPL	1
1.2	Almacenado y ejecución de programas	4
2	Variables globales	7
2.1	Uso de variables globales	7
2.2	Reordenación de variables	18
2.3	Soluciones a las actividades	23
3	Variables locales	24
3.1	Uso de variables locales	24
3.1.1	Estructura de variable local	24
3.1.2	Etiquetado de resultados	30
3.2	Programa con resultados múltiples	32
3.3	Uso de variables auxiliares	36
4	Presentación de resultados con MSGBOX	39
5	Ejecución condicionada	45
5.1	Estructura IF-THEN-ELSE	45
5.2	Estructura CASE	49
5.3	Soluciones de las actividades	53
6	Formularios de entrada de datos	54
6.1	Formularios de entrada de datos	54
6.2	Uso simplificado del comando INFORM	56
6.3	Un ejemplo de aplicación	60
6.4	Uso avanzado del comando INFORM	72
6.4.1	Valores de reset	72
6.4.2	Especificaciones de formato	75
6.4.3	Formatos de campo	78
7	Gestión de la memoria	84
7.1	Organización de la Memoria de la calculadora	84
7.2	Acceso a los objetos de los puertos	90
7.3	Copias de seguridad del sistema	92

1 Introducción

1.1 Programas en User-RPL

En una primera aproximación, podemos decir que un programa en User-RPL es una secuencia de objetos escritos entre los delimitadores << >>. Si cargamos un programa en la pila y pulsamos [EVAL], la calculadora procesa automáticamente la secuencia de objetos que contiene el programa. Es esencial entender que, al ejecutar el programa, la calculadora realizará diferentes acciones dependiendo del tipo de objeto.

- Comandos. Los comandos son ejecutados.
- Otros tipos de objetos son cargados en la pila, en concreto se cargan en la pila:
 - Expresiones algebraicas: 'cos(x)+sin(x)'.
 - Números: 2, 2.34, (1,2.34).
 - Strings: ''Entra un número'', ''Volumen''.
 - Listas: {1,2,'a','b'}.

Consideremos como ejemplo el siguiente programa.

```
DEG R&Z HEX R= 'X'
{HOME}
1:
2:
3:
4:
5:
6:
7:
8:
9:
10:
11:
12:
13:
14:
15:
16:
17:
18:
19:
20:
21:
22:
23:
24:
25:
26:
27:
28:
29:
30:
31:
32:
33:
34:
35:
36:
37:
38:
39:
40:
41:
42:
43:
44:
45:
46:
47:
48:
49:
50:
51:
52:
53:
54:
55:
56:
57:
58:
59:
60:
61:
62:
63:
64:
65:
66:
67:
68:
69:
70:
71:
72:
73:
74:
75:
76:
77:
78:
79:
80:
81:
82:
83:
84:
85:
86:
87:
88:
89:
90:
91:
92:
93:
94:
95:
96:
97:
98:
99:
100:
101:
102:
103:
104:
105:
106:
107:
108:
109:
110:
111:
112:
113:
114:
115:
116:
117:
118:
119:
120:
121:
122:
123:
124:
125:
126:
127:
128:
129:
130:
131:
132:
133:
134:
135:
136:
137:
138:
139:
140:
141:
142:
143:
144:
145:
146:
147:
148:
149:
150:
151:
152:
153:
154:
155:
156:
157:
158:
159:
160:
161:
162:
163:
164:
165:
166:
167:
168:
169:
170:
171:
172:
173:
174:
175:
176:
177:
178:
179:
180:
181:
182:
183:
184:
185:
186:
187:
188:
189:
190:
191:
192:
193:
194:
195:
196:
197:
198:
199:
200:
201:
202:
203:
204:
205:
206:
207:
208:
209:
210:
211:
212:
213:
214:
215:
216:
217:
218:
219:
220:
221:
222:
223:
224:
225:
226:
227:
228:
229:
230:
231:
232:
233:
234:
235:
236:
237:
238:
239:
240:
241:
242:
243:
244:
245:
246:
247:
248:
249:
250:
251:
252:
253:
254:
255:
256:
257:
258:
259:
260:
261:
262:
263:
264:
265:
266:
267:
268:
269:
270:
271:
272:
273:
274:
275:
276:
277:
278:
279:
280:
281:
282:
283:
284:
285:
286:
287:
288:
289:
290:
291:
292:
293:
294:
295:
296:
297:
298:
299:
300:
301:
302:
303:
304:
305:
306:
307:
308:
309:
310:
311:
312:
313:
314:
315:
316:
317:
318:
319:
320:
321:
322:
323:
324:
325:
326:
327:
328:
329:
330:
331:
332:
333:
334:
335:
336:
337:
338:
339:
340:
341:
342:
343:
344:
345:
346:
347:
348:
349:
350:
351:
352:
353:
354:
355:
356:
357:
358:
359:
360:
361:
362:
363:
364:
365:
366:
367:
368:
369:
370:
371:
372:
373:
374:
375:
376:
377:
378:
379:
380:
381:
382:
383:
384:
385:
386:
387:
388:
389:
390:
391:
392:
393:
394:
395:
396:
397:
398:
399:
400:
401:
402:
403:
404:
405:
406:
407:
408:
409:
410:
411:
412:
413:
414:
415:
416:
417:
418:
419:
420:
421:
422:
423:
424:
425:
426:
427:
428:
429:
430:
431:
432:
433:
434:
435:
436:
437:
438:
439:
440:
441:
442:
443:
444:
445:
446:
447:
448:
449:
450:
451:
452:
453:
454:
455:
456:
457:
458:
459:
460:
461:
462:
463:
464:
465:
466:
467:
468:
469:
470:
471:
472:
473:
474:
475:
476:
477:
478:
479:
480:
481:
482:
483:
484:
485:
486:
487:
488:
489:
490:
491:
492:
493:
494:
495:
496:
497:
498:
499:
500:
501:
502:
503:
504:
505:
506:
507:
508:
509:
510:
511:
512:
513:
514:
515:
516:
517:
518:
519:
520:
521:
522:
523:
524:
525:
526:
527:
528:
529:
530:
531:
532:
533:
534:
535:
536:
537:
538:
539:
540:
541:
542:
543:
544:
545:
546:
547:
548:
549:
550:
551:
552:
553:
554:
555:
556:
557:
558:
559:
560:
561:
562:
563:
564:
565:
566:
567:
568:
569:
570:
571:
572:
573:
574:
575:
576:
577:
578:
579:
580:
581:
582:
583:
584:
585:
586:
587:
588:
589:
590:
591:
592:
593:
594:
595:
596:
597:
598:
599:
600:
601:
602:
603:
604:
605:
606:
607:
608:
609:
610:
611:
612:
613:
614:
615:
616:
617:
618:
619:
620:
621:
622:
623:
624:
625:
626:
627:
628:
629:
630:
631:
632:
633:
634:
635:
636:
637:
638:
639:
640:
641:
642:
643:
644:
645:
646:
647:
648:
649:
650:
651:
652:
653:
654:
655:
656:
657:
658:
659:
660:
661:
662:
663:
664:
665:
666:
667:
668:
669:
670:
671:
672:
673:
674:
675:
676:
677:
678:
679:
680:
681:
682:
683:
684:
685:
686:
687:
688:
689:
690:
691:
692:
693:
694:
695:
696:
697:
698:
699:
700:
701:
702:
703:
704:
705:
706:
707:
708:
709:
710:
711:
712:
713:
714:
715:
716:
717:
718:
719:
720:
721:
722:
723:
724:
725:
726:
727:
728:
729:
730:
731:
732:
733:
734:
735:
736:
737:
738:
739:
740:
741:
742:
743:
744:
745:
746:
747:
748:
749:
750:
751:
752:
753:
754:
755:
756:
757:
758:
759:
760:
761:
762:
763:
764:
765:
766:
767:
768:
769:
770:
771:
772:
773:
774:
775:
776:
777:
778:
779:
780:
781:
782:
783:
784:
785:
786:
787:
788:
789:
790:
791:
792:
793:
794:
795:
796:
797:
798:
799:
800:
801:
802:
803:
804:
805:
806:
807:
808:
809:
810:
811:
812:
813:
814:
815:
816:
817:
818:
819:
820:
821:
822:
823:
824:
825:
826:
827:
828:
829:
830:
831:
832:
833:
834:
835:
836:
837:
838:
839:
840:
841:
842:
843:
844:
845:
846:
847:
848:
849:
850:
851:
852:
853:
854:
855:
856:
857:
858:
859:
860:
861:
862:
863:
864:
865:
866:
867:
868:
869:
870:
871:
872:
873:
874:
875:
876:
877:
878:
879:
880:
881:
882:
883:
884:
885:
886:
887:
888:
889:
890:
891:
892:
893:
894:
895:
896:
897:
898:
899:
900:
901:
902:
903:
904:
905:
906:
907:
908:
909:
910:
911:
912:
913:
914:
915:
916:
917:
918:
919:
920:
921:
922:
923:
924:
925:
926:
927:
928:
929:
930:
931:
932:
933:
934:
935:
936:
937:
938:
939:
940:
941:
942:
943:
944:
945:
946:
947:
948:
949:
950:
951:
952:
953:
954:
955:
956:
957:
958:
959:
960:
961:
962:
963:
964:
965:
966:
967:
968:
969:
970:
971:
972:
973:
974:
975:
976:
977:
978:
979:
980:
981:
982:
983:
984:
985:
986:
987:
988:
989:
990:
991:
992:
993:
994:
995:
996:
997:
998:
999:
1000:
1001:
1002:
1003:
1004:
1005:
1006:
1007:
1008:
1009:
1010:
1011:
1012:
1013:
1014:
1015:
1016:
1017:
1018:
1019:
1020:
1021:
1022:
1023:
1024:
1025:
1026:
1027:
1028:
1029:
1030:
1031:
1032:
1033:
1034:
1035:
1036:
1037:
1038:
1039:
1040:
1041:
1042:
1043:
1044:
1045:
1046:
1047:
1048:
1049:
1050:
1051:
1052:
1053:
1054:
1055:
1056:
1057:
1058:
1059:
1060:
1061:
1062:
1063:
1064:
1065:
1066:
1067:
1068:
1069:
1070:
1071:
1072:
1073:
1074:
1075:
1076:
1077:
1078:
1079:
1080:
1081:
1082:
1083:
1084:
1085:
1086:
1087:
1088:
1089:
1090:
1091:
1092:
1093:
1094:
1095:
1096:
1097:
1098:
1099:
1100:
1101:
1102:
1103:
1104:
1105:
1106:
1107:
1108:
1109:
1110:
1111:
1112:
1113:
1114:
1115:
1116:
1117:
1118:
1119:
1120:
1121:
1122:
1123:
1124:
1125:
1126:
1127:
1128:
1129:
1130:
1131:
1132:
1133:
1134:
1135:
1136:
1137:
1138:
1139:
1140:
1141:
1142:
1143:
1144:
1145:
1146:
1147:
1148:
1149:
1150:
1151:
1152:
1153:
1154:
1155:
1156:
1157:
1158:
1159:
1160:
1161:
1162:
1163:
1164:
1165:
1166:
1167:
1168:
1169:
1170:
1171:
1172:
1173:
1174:
1175:
1176:
1177:
1178:
1179:
1180:
1181:
1182:
1183:
1184:
1185:
1186:
1187:
1188:
1189:
1190:
1191:
1192:
1193:
1194:
1195:
1196:
1197:
1198:
1199:
1200:
1201:
1202:
1203:
1204:
1205:
1206:
1207:
1208:
1209:
1210:
1211:
1212:
1213:
1214:
1215:
1216:
1217:
1218:
1219:
1220:
1221:
1222:
1223:
1224:
1225:
1226:
1227:
1228:
1229:
1230:
1231:
1232:
1233:
1234:
1235:
1236:
1237:
1238:
1239:
1240:
1241:
1242:
1243:
1244:
1245:
1246:
1247:
1248:
1249:
1250:
1251:
1252:
1253:
1254:
1255:
1256:
1257:
1258:
1259:
1260:
1261:
1262:
1263:
1264:
1265:
1266:
1267:
1268:
1269:
1270:
1271:
1272:
1273:
1274:
1275:
1276:
1277:
1278:
1279:
1280:
1281:
1282:
1283:
1284:
1285:
1286:
1287:
1288:
1289:
1290:
1291:
1292:
1293:
1294:
1295:
1296:
1297:
1298:
1299:
1300:
1301:
1302:
1303:
1304:
1305:
1306:
1307:
1308:
1309:
1310:
1311:
1312:
1313:
1314:
1315:
1316:
1317:
1318:
1319:
1320:
1321:
1322:
1323:
1324:
1325:
1326:
1327:
1328:
1329:
1330:
1331:
1332:
1333:
1334:
1335:
1336:
1337:
1338:
1339:
1340:
1341:
1342:
1343:
1344:
1345:
1346:
1347:
1348:
1349:
1350:
1351:
1352:
1353:
1354:
1355:
1356:
1357:
1358:
1359:
1360:
1361:
1362:
1363:
1364:
1365:
1366:
1367:
1368:
1369:
1370:
1371:
1372:
1373:
1374:
1375:
1376:
1377:
1378:
1379:
1380:
1381:
1382:
1383:
1384:
1385:
1386:
1387:
1388:
1389:
1390:
1391:
1392:
1393:
1394:
1395:
1396:
1397:
1398:
1399:
1400:
1401:
1402:
1403:
1404:
1405:
1406:
1407:
1408:
1409:
1410:
1411:
1412:
1413:
1414:
1415:
1416:
1417:
1418:
1419:
1420:
1421:
1422:
1423:
1424:
1425:
1426:
1427:
1428:
1429:
1430:
1431:
1432:
1433:
1434:
1435:
1436:
1437:
1438:
1439:
1440:
1441:
1442:
1443:
1444:
1445:
1446:
1447:
1448:
1449:
1450:
1451:
1452:
1453:
1454:
1455:
1456:
1457:
1458:
1459:
1460:
1461:
1462:
1463:
1464:
1465:
1466:
1467:
1468:
1469:
1470:
1471:
1472:
1473:
1474:
1475:
1476:
1477:
1478:
1479:
1480:
1481:
1482:
1483:
1484:
1485:
1486:
1487:
1488:
1489:
1490:
1491:
1492:
1493:
1494:
1495:
1496:
1497:
1498:
1499:
1500:
1501:
1502:
1503:
1504:
1505:
1506:
1507:
1508:
1509:
1510:
1511:
1512:
1513:
1514:
1515:
1516:
1517:
1518:
1519:
1520:
1521:
1522:
1523:
1524:
1525:
1526:
1527:
1528:
1529:
1530:
1531:
1532:
1533:
1534:
1535:
1536:
1537:
1538:
1539:
1540:
1541:
1542:
1543:
1544:
1545:
1546:
1547:
1548:
1549:
1550:
1551:
1552:
1553:
1554:
1555:
1556:
1557:
1558:
1559:
1560:
1561:
1562:
1563:
1564:
1565:
1566:
1567:
1568:
1569:
1570:
1571:
1572:
1573:
1574:
1575:
1576:
1577:
1578:
1579:
1580:
1581:
1582:
1583:
1584:
1585:
1586:
1587:
1588:
1589:
1590:
1591:
1592:
1593:
1594:
1595:
1596:
1597:
1598:
1599:
1600:
1601:
1602:
1603:
1604:
1605:
1606:
1607:
1608:
1609:
1610:
1611:
1612:
1613:
1614:
1615:
1616:
1617:
1618:
1619:
1620:
1621:
1622:
1623:
1624:
1625:
1626:
1627:
1628:
1629:
1630:
1631:
1632:
1633:
1634:
1635:
1636:
1637:
1638:
1639:
1640:
1641:
1642:
1643:
1644:
1645:
1646:
1647:
1648:
1649:
1650:
1651:
1652:
1653:
1654:
1655:
1656:
1657:
1658:
1659:
1660:
1661:
1662:
1663:
1664:
1665:
1666:
1667:
1668:
1669:
1670:
1671:
1672:
1673:
1674:
1675:
1676:
1677:
1678:
1679:
1680:
1681:
1682:
1683:
1684:
1685:
1686:
1687:
1688:
1689:
1690:
1691:
1692:
1693:
1694:
1695:
1696:
1697:
1698:
1699:
1700:
1701:
1702:
1703:
1704:
1705:
1706:
1707:
1708:
1709:
1710:
1711:
1712:
1713:
1714:
1715:
1716:
1717:
1718:
1719:
1720:
1721:
1722:
1723:
1724:
1725:
1726:
1727:
1728:
1729:
1730:
1731:
1732:
1733:
1734:
1735:
1736:
1737:
1738:
1739:
1740:
1741:
1742:
1743:
1744:
1745:
1746:
1747:
1748:
1749:
1750:
1751:
1752:
1753:
1754:
1755:
1756:
1757:
1758:
1759:
1760:
1761:
1762:
1763:
1764:
1765:
1766:
1767:
1768:
1769:
1770:
1771:
1772:
1773:
1774:
1775:
1776:
1777:
1778:
1779:
1780:
1781:
1782:
1783:
1784:
1785:
1786:
1787:
1788:
1789:
1790:
1791:
1792:
1793:
1794:
1795:
1796:
1797:
1798:
1799:
1800:
1801:
1802:
1803:
1804:
1805:
1806:
1807:
1808:
1809:
1810:
1811:
1812:
1813:
1814:
1815:
1816:
1817:
1818:
1819:
1820:
1821:
1822:
1823:
1824:
1825:
1826:
1827:
1828:
1829:
1830:
1831:
1832:
1833:
1834:
1835:
1836:
1837:
1838:
1839:
1840:
1841:
1842:
1843:
1844:
1845:
1846:
1847:
1848:
1849:
1850:
1851:
1852:
1853:
1854:
1855:
1856:
1857:
1858:
1859:
1860:
1861:
1862:
1863:
1864:
1865:
1866:
1867:
1868:
1869:
1870:
1871:
1872:
1873:
1874:
1875:
1876:
1877:
1878:
1879:
1880:
1881:
1882:
1883:
1884:
1885:
1886:
1887:
1888:
1889:
1890:
1891:
1892:
1893:
1894:
1895:
1896:
1897:
1898:
1899:
1900:
1901:
1902:
1903:
1904:
1905:
1906:
1907:
1908:
1909:
1910:
1911:
1912:
1913:
1914:
1915:
1916:
1917:
1918:
1919:
1920:
1921:
1922:
1923:
1924:
1925:
1926:
1927:
1928:
1929:
1930:
1931:
1932:
1933:
1934:
1935:
1936:
1937:
1938:
1939:
1940:
1941:
1942:
1943:
1944:
1945:
1946:
1947:
1948:
1949:
1950:
1951:
1952:
1953:
1954:
1955:
1956:
1957:
1958:
1959:
1960:
1961:
1962:
1963:
1964:
1965:
1966:
1967:
1968:
1969:
1970:
1971:
1972:
1973:
1974:
1975:
1976:
1977:
1978:
1979:
1980:
1981:
1982:
1983:
1984:
1985:
1986:
1987:
1988:
1989:
1990:
1991:
1992:
1993:
1994:
1995:
1996:
1997:
1998:
1999:
2000:
2001:
2002:
2003:
2004:
2005:
2006:
2007:
2008:
2009:
2010:
2011:
2012:
2013:
2014:
2015:
2016:
2017:
2018:
2019:
2020:
2021:
2022:
2023:
2024:
2025:
2026:
2027:
2028:
2029:
2030:
2031:
2032:
2033:
2034:
2035:
2036:
2037:
2038:
2039:
2040:
2041:
2042:
2043:
2044:
2045:
2046:
2047:
2048:
2049:
2050:
2051:
2052:
2053:
2054:
2055:
2056:
2057:
2058:
2059:
2060:
2061:
2062:
2063:
2064:
2065:
2066:
2067:
2068:
2069:
2070:
2071:
2072:
2073:
2074:
2075:
2076:
2077:
2078:
2079:
2080:
2081:
2082:
2083:
2084:
2085:
2086:
2087:
2088:
2089:
2090:
2091:
2092:
2093:
2094:
2095:
2096:
2097:
2098:
2099:
2100:
2101:
2102:
2103:
2104:
2105:
2106:
2107:
2108:
2109:
2110:
2111:
2112:
2113:
2114:
2115:
2116:
2117:
2118:
2119:
2120:
2121:
2122:
2123:
2124:
2125:
2126:
2127:
2128:
2129:
2130:
2131:
2132:
2133:
2134:
2135:
2136:
2137:
2138:
2139:
2140:
2141:
2142:
2143:
2144:
2145:
2146:
2147:
2148:
2149:
2150:
2151:
```

```

DEG R&Z HEX R= 'X'
{HOME}
4:
3:
2:
1:
1/2
EDIT | VIEW | STACK | RCL | PURGE | CLEAR

```

Al escribir programas en User-RPL, podemos tomar objetos de la pila y usar los comandos de pila. La siguiente tabla contiene algunos comandos útiles que permiten la manipulación de pila.

Comando	Acción	Sintaxis
DUP	Duplica el objeto del nivel 1 de la pila.	
SWAP	Intercambia el contenido de nivel 1 y el nivel 2 de la pila.	
DROP	Elimina el objeto del nivel 1 de la pila.	
DROPN	Elimina los objetos de los n primeros niveles de la pila.	n DROPN
DUPN	Duplica los objetos de los n primeros niveles de la pila.	n DUPN
DUP2	Duplica el contenido de los 2 primeros niveles de la pila.	

Puedes obtener los comandos de pila en el menú [PRG][STACK].

```

DEG R&Z HEX R= 'X'
{HOME}
4:
3:
2:
1:
1/2
DUP | SWAP | DROP | OVER | ROT | UNROT

```

También puedes obtenerlos en catálogo de comandos.

```

DEG CATALOG: 762 COMMANDS
{HOME}
4:
3:
2:
1:
D-TAG
DUP
DUP2
DUPDUP
DUPN
D-R
CANCL | OK

```

Actividad 1.1 El menú [PRG] contiene todos los comandos útiles para programación agrupados por categorías.


```

DEG R&Z HEX R= 'X'
{HOME}
5:
4:
3:
2:
1:
MED CASDI

```

observamos que el nombre MED aparece asociado a la tecla [F1]. Para ejecutar el programa, simplemente tenemos que cargar dos números en la pila, por ejemplo 4 y 8

```

DEG R&Z HEX R= 'X'
{HOME}
5:
4:
3:
2:
1:
MED CASDI

```

4

y pulsar [F1], obtendremos

```

DEG R&Z HEX R= 'X'
{HOME}
5:
4:
3:
2:
1:
MED CASDI

```

6

Si ahora pulsamos nuevamente la tecla [F1], obtenemos un error.

```

DEG R&Z HEX R= 'X'
{HOME}
5:
4:
3:
2:
1:
MED CASDI

```

⚠ + Error:
Too Few
Arguments

6

El error se produce porque el programa necesita dos valores en la pila para funcionar correctamente.

Cargamos los valores 4 y 5 en la pila

```

DEG R4Z HEX R= 'X'
{HOME}
4:
3:
2:
1:
MED CASDI

```

y escribimos el nombre del programa

```

DEG R4Z HEX R= 'X'
{HOME}
4:
3:
2:
1:
MED
MED CASDI

```

al pulsar [ENTER], como el nombre MED corresponde a un programa, se produce la evaluación automática y obtenemos

```

DEG R4Z HEX R= 'X'
{HOME}
4:
3:
2:
1:
MED CASDI

```

Si deseamos cargar el nombre del programa sin que se ejecute, debemos escribirlo ente comillas simples.

```

DEG R4Z HEX R= 'X'
{HOME}
4:
3:
2:
1:
'MED'
MED CASDI

```

Entonces al pulsar [EVAL] se evalúa el nombre y se produce la ejecución del programa.

```

DEG R&Z HEX R= 'X'
[HOME]
44:
50:
52:
1:
2/2
MED CASDI

```

Resumiendo, supongamos que hemos guardado un programa con el nombre NOMPRG, entonces el nombre NOMPRG pasa a comportarse como cualquier otro comando del sistema. Podemos ejecutarlo directamente usando la tecla de función correspondiente en el área de variables, o bien usar su nombre.

2 Variables globales

Los programas pueden manejar variables. Las *variables globales*

- Son variables ordinarias.
- Su nombre aparece en el directorio de trabajo.
- Persisten una vez ejecutado el programa, salvo que el programa las borre explícitamente.

En contrapartida, veremos en la sección siguiente que existen otro tipo de variables: *las variables locales*. Las variables locales sólo se pueden usar dentro del programa que las crea, no aparecen en el directorio de trabajo y no interfieren con otras variables previamente existentes.

2.1 Uso de variables globales

Para asignar valores a una variable se usa el comando STO, dado que una variable global puede existir previamente a la ejecución del programa, debes usar su nombre entre apóstrofes (*quoted name*).

Como ejemplo, vamos a realizar un programa que calcule el volumen, área lateral y área total de un cilindro a partir del radio y la altura. Con mayor detalle, creamos un subdirectorio de HOME denominado CILIN, el programa se denominará VACL(Volumen y Area del CiLindro). Tomará los valores:

- Radio: almacenado en la variable global R.
- Altura: almacenado en la variable global H.

Y producirá como salida los valores:

- Volumen: almacenado en la variable global Vol.
- Area de las bases: almacenado en la variable global Abas.
- Area lateral: almacenado en la variable global Alat.
- Area total: almacenado en la variable global Atot.

Emplearemos las fórmulas:

- Volumen: $V = \pi R^2 H$.
- Area de las bases: $A_{bas} = 2\pi R^2$.
- Area lateral: $A_{lat} = 2\pi RH$.
- Area global: $A_{tot} = A_{bas} + A_{lat}$.

Como valores iniciales y valores de prueba, usaremos

R	H	Vol	$Abas$	$Alat$	$Atot$
3	10	282.7433	56.5486	188.4956	245.0442

(1)

Actividad 2.1 Realiza los siguientes pasos.

1. En primer lugar, sitúate en el directorio HOME y crea el subdirectorio CILIN.

```
DEG R&2 HEX R= 'X'
{HOME}
0:
4:
8:
C:
1: 'CILIN'
PURGE RCL STO TPA CRDIR PGDIR
```

2. Una vez creado, entra en el directorio CILIN.

```
DEG R&2 HEX R= 'X'
{HOME CILIN}
```

3. Crea la variable R con el valor inicial 3, para ello escribe

```
DEG R&Z HEX R= 'X'
{HOME CILIN}
5:
4:
3:
2:
1: 'R' 3
EDIT VIEW STACK RCL PURGE CLEAR
```

y pulsa [STO]. De forma análoga, crea la variable H con el valor inicial 10, pulsa [VAR] para observar las variables creadas.

```
DEG R&Z HEX R= 'X'
{HOME CILIN}
5:
4:
3:
2:
1:
H R
```

4. Accede al editor de ecuaciones y escribe la expresión algebraica correspondiente al volumen.

```

          2
      πR H
H R
```

Si pulsas [VAR] cuando estás dentro del editor de ecuaciones podrás usar las teclas de función para escribir las variables H y R. Una vez completada la fórmula, pulsa [ENTER] para cargar la fórmula en la pila.

```
DEG R&Z HEX R= 'X'
{HOME CILIN}
5:
4:
3:
2:
1: π·R2·H
H R
```

5. Escribe y carga en la pila las restantes fórmulas.

```

DEG R42 HEX R= 'X'
{HOME CILIN}
-----
3:
2:
1:
H R

```

6. Ahora vamos a escribir el programa, pulsa ↵[+] para escribir los delimitadores del programa.

```

DEG R42 HEX R= 'X'
{HOME CILIN}
-----
2:
1:
*
*
H R

```

Pulsa [HIST] para acceder al editor de pila y copia con ECHO la fórmula correspondiente al volumen.

```

DEG R42 HEX R= 'X'
{HOME CILIN}
-----
5:
4:
3:
2:
1:
ECHO VIEW EDIT PICK ROLL ROLLO

```

Pulsa [ENTER] para volver a la edición del programa.

```

DEG R42 HEX R= 'X'
{HOME CILIN}
-----
2:
1:
* 'π*R^2*H'
*
H R

```

7. Observa que cuando te encuentras dentro de los delimitadores de programa, se activa el modo PRG. Cuando el modo PRG está activo, las teclas escriben los comandos y funciones correspondientes, en vez de producir su ejecución.

```

DEG R&2 HEX R= 'X'
{HOME CILIN}
... 2
2: 2·π·R2
1: 2·π·R·H
« 'π·R2·H'
»
H | R | | | |

```

Pulsa la tecla [EVAL] para escribir el comando EVAL, escribe el nombre de variable 'Vol' (no olvides los apóstrofes) y pulsa la tecla [STO] para escribir el comando STO.

```

DEG R&2 HEX R= 'X'
{HOME CILIN}
... 2
2: 2·π·R2
1: 2·π·R·H
...2·H' EVAL 'Vol' STO
...
H | R | | | |

```

8. Pulsa [ENTER], para cargar el programa en la pila.

```

DEG R&2 HEX R= 'X'
{HOME CILIN}
... 2
3: 2·π·R2
2: 2·π·R·H
1: « 'π·R2·H' EVAL
'Vol' STO »
H | R | | | |

```

9. Escribe el nombre del programa VACL

```

DEG R&2 HEX R= 'X'
{HOME CILIN}
... 2
4: 2·π·R2
3: 2·π·R·H
2: « 'π·R2·H' EVAL
'Vol' STO »
1: 'VACL'
H | R | | | |

```

y pulsa [STO].

```

DEG R&2 HEX R= 'X'
{HOME CILIN}
T:
3:
2:
1:
          π·R2·H
          2·π·R2
          2·π·R·H
VACL | H | R |

```

10. Vamos a probar el funcionamiento de la primera parte del programa, pulsa [F1], obtendrás:

```

DEG R&2 HEX R= 'X'
{HOME CILIN}
T:
3:
2:
1:
          π·R2·H
          2·π·R2
          2·π·R·H
Vol | VACL | H | R |

```

Observa que el programa ha creado la variable Vol, si pulsas [F1], obtendrás el valor de Vol.

```

DEG R&2 HEX R= 'X'
{HOME CILIN}
4: ▲
3:
2:
1:
          π·R2·H
          2·π·R2
          2·π·R·H
          90·π
Vol | VACL | H | R |

```

Como que la calculadora está en modo exacto, hemos obtenido 90π , pulsa ↑[ENTER] para ejecutar → NUM, obtendrás:

```

DEG R&2 HEX R= 'X'
{HOME CILIN}
4: ▲
3:
2:
1:
          π·R2·H
          2·π·R2
          2·π·R·H
          282.743338823
STD | FIX | SCI | ENG | FN, | NL |

```

que es el valor correspondiente al volumen en la tabla (1) de valores iniciales y de prueba.

11. Antes de continuar, fija la calculadora en modo aproximado y el formato numérico en FIX 4.

```

DEG R↔Z HEX R↔ 'X'
{HOME CILIND}
4:      ▲
3:      π·R2·H
2:      2·π·R2
1:      2·π·R·H
          282.7433
STD | FIX | SCI | ENG | FM, | NL |

```

12. Pulsa [VAR] para acceder al menú de variables del directorio

```

DEG R↔Z HEX R↔ 'X'
{HOME CILIND}
4:      ▲
3:      π·R2·H
2:      2·π·R2
1:      2·π·R·H
          282.7433
Vol | VACL | H | R |

```

y pulsa ↵[F2] para recuperar el programa.

```

DEG R↔Z HEX R↔ 'X'
{HOME CILIND}
4:      ▲
3:      2·π·R2
2:      2·π·R·H
1:      282.7433
     « 'π·R2·H' EVAL
     'Vol' STO »
Vol | VACL | H | R |

```

13. Pulsa [▼] para activar el editor.

```

DEG R↔Z HEX R↔ 'X'
{HOME CILIND}
PRG
« 'π·R2·H' EVAL
'Vol' STO
»
←SKIP | SKIP→ | +DEL | DEL← | DEL→ | L | INS |

```

14. La tecla \uparrow (10,3) permite saltar de línea,



la usaremos para escribir con más comodidad el programa¹.

```
DEG R42 HEX R~ 'X' PRG
{HOME CILIND}
* 'π*R^2*H' EVAL
'Vol1' STO
*
*
+SKIP|SKIP+ +DEL|DEL+|DEL L|INS
```

15. Pulsa [HIST] para copiar la fórmula correspondiente a la suma de áreas de las bases y completa el programa como sigue

```
DEG R42 HEX R~ 'X' PRG
{HOME CILIND}
* 'π*R^2*H' EVAL
'Vol1' STO
'2*π*R^2' EVAL
'Abas' STO *
*
+SKIP|SKIP+ +DEL|DEL+|DEL L|INS
```

16. A continuación copia la fórmula del área lateral y escribe la parte correspondiente de programa².

```
DEG R42 HEX R~ 'X' PRG
{HOME CILIND}
'2*π*R^2' EVAL
'Abas' STO
'2*π*R*H' EVAL
'Alat' STO
*
+SKIP|SKIP+ +DEL|DEL+|DEL L|INS
```

¹Los saltos de línea desaparecen al cargar el programa en la pila.

²Observa que los nombres de las variables Abas y Alat deben estar entre apóstrofes (quoted names) en caso contrario, se produciría un error en ejecuciones del programa posteriores a la primera.

17. Vamos a probar el funcionamiento del programa, pulsa [ENTER] para cargar el programa en la pila.

```
DEG R&2 HEX R~ 'X'
{HOME CILIND}
1: « 'π·R^2.0000·H'
  EVAL 'Vol' STO '
  2.0000·π·R^2.0000'
  EVAL 'Abas' STO '
  2.0000·π·R·H' EVAL
Vol | VACL | H | R |
```

Pulsa \uparrow [F2] para almacenarlo con el nombre VACL, sustituyendo a la versión anterior.

```
DEG R&2 HEX R~ 'X'
{HOME CILIND}
4:      ▲      π·R^2·H
3:      2·π·R^2
2:      2·π·R·H
1:      282.7433
Vol | VACL | H | R |
```

Pulsa ahora [F2] para ejecutar la versión actualizada del programa

```
DEG R&2 HEX R~ 'X'
{HOME CILIND}
4:      ▲      π·R^2·H
3:      2·π·R^2
2:      2·π·R·H
1:      282.7433
Alat | Abas | Vol | VACL | H | R
```

observa que el programa ha creado las variables Alat y Abas. Pulsa [F1] y [F2] para obtener los valores que contienen las variables.

```
DEG R&2 HEX R~ 'X'
{HOME CILIND}
5:      ▲      2·π·R^2
4:      2·π·R·H
3:      282.7433
2:      188.4956
1:      56.5487
Alat | Abas | Vol | VACL | H | R
```

Compara los valores obtenidos con los que aparecen en la Tabla 1 de la página 8.

18. Vamos a terminar. Primer pulsa [CLEAR] para limpiar³ la pila.

```
DEG R&Z HEX R~ 'X'
{HOME CILIN}
5:
4:
3:
2:
1:
Alat Abas Vol VACL H R
```

Pulsa ⤴[F4] para recuperar el programa guardado en [VACL].

```
DEG R&Z HEX R~ 'X'
{HOME CILIN}
1: « 'π*R^2.0000*H'
  EVAL 'Vol' STO '
  2.0000*π*R^2.0000'
  EVAL 'Abas' STO '
  2.0000*π*R*H' EVAL
Alat Abas Vol VACL H R
```

Pulsa [▼] para activar el editor.

```
DEG R&Z HEX R~ 'X' PRG
{HOME CILIN}
'Vol' STO '2.*π*R^2.'
EVAL 'Abas' STO '2.*π
*R*H' EVAL 'Alat' STO
♦
⌘
+SKIP|SKIP+|+DEL|DEL+|DEL L|INS ▢
```

Para acabar el programa, escribe los nombre Alat y Abas sin apóstrofes,

```
DEG R&Z HEX R~ 'X' PRG
{HOME CILIN}
'Vol' STO '2.*π*R^2.'
EVAL 'Abas' STO '2.*π
*R*H' EVAL 'Alat' STO
Alat Abas ♦
⌘
Alat Abas Vol VACL H R
```

³Tecla ⤴(4,5).

eso hace que se carguen sus valores en la pila, pulsa [+] para entrar el comando +

```
DEG R42 HEX R~ 'X' PRG
{HOME CILIN}
'Vol' STO '2.*π*R^2.'
EVAL 'Abas' STO '2.*π
*R*H' EVAL 'Alat' STO
Alat Abas +
⊗
Alat | Abas | Vol | VACL | H | R
```

19. Ahora tendrás el valor $Alat + Abas$ en la pila, entra el nombre $Atot$ entre apóstrofos⁴ y escribe STO.


```
DEG R42 HEX R~ 'X' PRG
{HOME CILIN}
'Vol' STO '2.*π*R^2.'
EVAL 'Abas' STO '2.*π
*R*H' EVAL 'Alat' STO
Alat Abas +
'Atot' STO
Alat | Abas | Vol | VACL | H | R
```

Pulsa [ENTER] para cargar el programa en la pila

```
DEG R42 HEX R~ 'X' PRG
{HOME CILIN}
1: ⋈ 'π*R^2.0000*H'
EVAL 'Vol' STO '
2.0000*π*R^2.0000'
EVAL 'Abas' STO '
2.0000*π*R*H' EVAL
Alat | Abas | Vol | VACL | H | R
```

y pulsa \uparrow [F4] para guardarlo con el nombre VACL.

```
DEG R42 HEX R~ 'X' PRG
{HOME CILIN}
5:
4:
3:
2:
1:
Alat | Abas | Vol | VACL | H | R
```



⁴Observa que en este caso quieres usar el nombre, no el valor, por eso debes escribirlo entre apóstrofos.

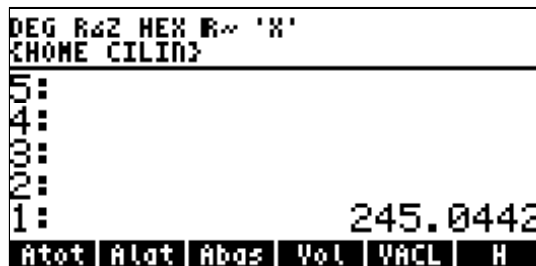
Pulsa [F4] para ejecutar el programa.

```
DEG R&Z HEX R~ 'X'
{HOME CILIN}
0:
4:
00:
0:
1:
Atot | Alat | Abas | Vol | VACL | H
```



Observa que el programa ha creado la variable Atot, pulsa [F1] para obtener su valor.

```
DEG R&Z HEX R~ 'X'
{HOME CILIN}
0:
4:
00:
0:
1: 245.0442
Atot | Alat | Abas | Vol | VACL | H
```



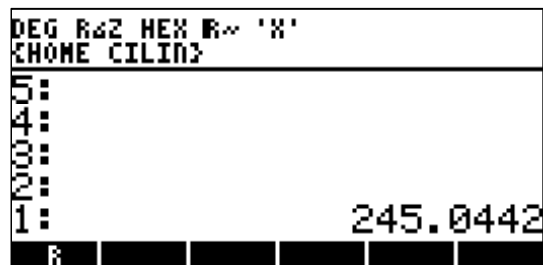
2.2 Reordenación de variables

En la actividad anterior hemos observado que las nuevas variables se van añadiendo por la izquierda conforme se crean, en tanto que las variables antiguas se desplazan a la derecha. Esto ha motivado que la variable *R* haya pasado a la segunda página del menú

```
DEG R&Z HEX R~ 'X'
{HOME CILIN}
0:
4:
00:
0:
1: 245.0442
Atot | Alat | Abas | Vol | VACL | H
```



```
DEG R&Z HEX R~ 'X'
{HOME CILIN}
0:
4:
00:
0:
1: 245.0442
R | | | | |
```



Es posible que queramos tener otra ordenación de las variables. En nuestro ejemplo, una ordenación más conveniente podría ser la siguiente:

[H] [R] [VACL] [Vol] [Atot] [Alat] [Abas]

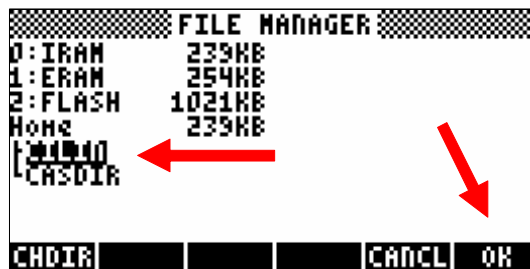
De esta forma, aparecerían en primer lugar las teclas correspondientes a las variables R y H que contienen los datos de entrada, seguidamente, tendríamos la tecla correspondiente al programa y, a continuación, las teclas correspondientes a las variables de salida, ordenadas de manera que la variable Abas (área de las bases) pasaría a la segunda página del menú. La siguiente actividad muestra como reorganizar el contenido de un directorio usando la aplicación de gestión de variables y directorios FILES.

Actividad 2.2 Para reorganizar el directorio, realiza los siguientes pasos:

1. Pulsa [FILES]



para acceder a la aplicación de gestión de variables y directorios, verás que el directorio actual está resaltado.



Pulsa [ENTER] o [OK] para entrar en el contenido del directorio

Memory: 244377 Select: 0	
DR Atot	REAL 10
DR Alat	REAL 10
DR Abas	REAL 10
DR Vol	REAL 10
DR VACL	PROG 171
DR H	INTG 6
DR R	INTG 6

EDIT COPY MOVE RCL EVAL TREE

2. Usa las teclas de desplazamiento para resaltar la variable H

Memory: 244377 Select: 0	
DR Atot	REAL 10
DR Alat	REAL 10
DR Abas	REAL 10
DR Vol	REAL 10
DR VACL	PROG 171
DR H	INTG 6
DR R	INTG 6
EDIT COPY MOVE RCL EVAL TREE	

y pulsa [ENTER].

Memory: 244377 Select: 6	
DR Atot	REAL 10
DR Alat	REAL 10
DR Abas	REAL 10
DR Vol	REAL 10
DR VACL	PROG 171
1 DR H	INTG 6
DR R	INTG 6
EDIT COPY MOVE RCL EVAL TREE	

Observa que ha aparecido un 1 junto a la variable H y se ha resaltado la variable R.

3. Pulsa [ENTER] para asignar el número de orden 2 a la variable R.

Memory: 244377 Select: 12	
DR Atot	REAL 10
DR Alat	REAL 10
DR Abas	REAL 10
DR Vol	REAL 10
DR VACL	PROG 171
1 DR H	INTG 6
2 DR R	INTG 6
EDIT COPY MOVE RCL EVAL TREE	

4. Resalta el programa VACL y pulsa [ENTER] para asignarle el número de orden 3.


Memory: 244377 Select: 184	
DR Atot	REAL 10
DR Alat	REAL 10
DR Abas	REAL 10
DR Vol	REAL 10
3 DR VACL	PROG 171
1 DR H	INTG 6
2 DR R	INTG 6
EDIT COPY MOVE RCL EVAL TREE	

5. Continúa asignando números de orden hasta que obtengas la siguiente pantalla.

Memory: 244377 Select: 226	
5 DR Atot	REAL 10
6 DR Alot	REAL 10
7 DR Ados	REAL 10
4 DR Vol	REAL 10
3 DR VACL	PROG 171
1 DR H	INTG 6
2 DR R	INTG 6
EDIT COPY MOVE RCL EVAL TREE	

6. Ahora pulsa [NEXT] para pasar a la segunda página de menú

Memory: 244377 Select: 226	
5 DR Atot	REAL 10
6 DR Alot	REAL 10
7 DR Ados	REAL 10
4 DR Vol	REAL 10
3 DR VACL	PROG 171
1 DR H	INTG 6
2 DR R	INTG 6
PURGE RENAM NEW ORDER SEND RECV	



- y pulsa [F4] para ejecutar ORDER.

Memory: 243920 Select: 0	
DR H	INTG 6
DR R	INTG 6
DR VACL	PROG 171
DR Vol	REAL 10
DR Atot	REAL 10
DR Alot	REAL 10
DR Ados	REAL 10
PURGE RENAM NEW ORDER SEND RECV	

7. Pulsa [CANCEL] para salir de FILES. Observa que el directorio ha quedado organizado de la forma deseada.

DEG R22 HEX R~ 'X'					
[HOME] CILIN3					
0:					
4:					
0:					
0:					
1:				245.0442	
H	R	VACL	Vol	Atot	Alot

```

DEG R&Z HEX R~ 'X'
{HOME CILIN}
50:
40:
30:
20:
1: 245.0442
Abas

```

Actividad 2.3 *Completa la siguiente tabla usando el programa VACL*

R	H	Vol	Abas	Alat	Atot
3.5	12				

Actividad 2.4 *Crea un directorio con el nombre ESFE. Escribe un programa con el nombre CVAE (Cálculo del Volumen y Area de la Esfera) que calcule el volumen y el área de la superficie esférica a partir del radio. La entrada es una variable global R que contiene el radio y la salida son las variables globales Vol y Area. Usa las fórmulas*

$$V = \frac{4}{3}\pi R^3, \quad A = 4\pi R^2.$$

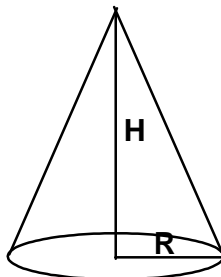
Usa la siguiente tabla de valores iniciales y de valores de verificación

R	H	Vol	Area
3.5	12.1	179.59	153.99

Actividad 2.5 *Reorganiza el directorio ESFE de forma que las variables aparezcan en el siguiente orden*

[R] [CVAE] [Vol] [Area].

Actividad 2.6 *Crea un directorio con el nombre CONO. Escribe un programa con el nombre CVAC (Cálculo del Volumen y Area del cono) que calcule el volumen, el área lateral, el área de la base y el área total de la superficie cónica a partir del radio de la base y la altura.*



La entrada son las siguientes variables globales:

- R, que contiene valor del radio.
- H, que contiene el valor de la altura.

La salida está formada por las siguientes variables globales:

- Vol, que contiene el valor del volumen.
- Abas, que contiene el valor del área de la base.
- Alat, que contiene el valor del área lateral.
- Atot, que contiene el valor del área total.

Usa las fórmulas

$$V = \frac{1}{3}\pi R^2 H, \quad A_{bas} = \pi R^2, \quad A_{lat} = \pi R \sqrt{R^2 + H^2}, \quad A_{tot} = A_{abs} + A_{lat},$$

y la siguiente tabla de valores iniciales y de valores de verificación

R	H	Vol	Abas	Alat	Atot
2.5	11.5	75.3	19.6	92.4	112.1

2.3 Soluciones a las actividades

Actividad 2.3

R	H	Vol	Abas	Alat	Atot
3.5	12.1	465.7	77.0	266.1	343.1

Actividad 2.4

Programa	Acción
<<	Abre programa.
'1/3*π*R^2*H' EVAL	Calcula el valor del volumen.
'Vol' STO	Guarda el valor con el nombre Vol.
'4*π*R^2' EVAL	Calcula el valor del área de la superficie esférica.
'Area' STO	Guarda el valor con el nombre Area.
>>	Cierra programa.

Actividad 2.6

Programa	Acción
<<	Abre programa.
'1/3*π*R^2*H' EVAL	Calcula el valor del volumen.
'Vol' STO	Guarda el valor con el nombre Vol.
'π*R^2' EVAL	Calcula el valor del área de la base.
'Abas' STO	Guarda el valor con el nombre Abas.
'π*R*√(R^2+H^2)' EVAL	Calcula el valor del área lateral.
'Alat' STO	Guarda el valor con el nombre Alat.
'Abas + Alat' EVAL	Calcula el valor del área total.
'Atot' STO	Guarda el valor con el nombre Atot.
>>	cCierra programa.

3 Variables locales

Las *variable locales* son un tipo de variable especialmente diseñado para su uso en programación. El comportamiento de las variables locales en User-RPL es similar al comportamiento que tienen las variables locales en otros lenguajes de programación: Pascal, C, Fortran, etc. La idea fundamental es que una variable local solo existe dentro del procedimiento o programa que la crea, no interacciona con variables globales del mismo nombre que puedan existir previamente y desaparecen automáticamente⁵ cuando la ejecución del programa finaliza.

En el caso del lenguaje de programación User-RPL, las variables locales tienen las siguientes características:

- No aparecen en el área de variables del directorio actual.
- Sólo son accesibles dentro del programa que las crea.
- No interaccionan con variables globales.
- El tiempo de acceso⁶ es inferior al tiempo de acceso de las variables globales.
- Las variables locales son borradas automáticamente al finalizar o interrumpirse el programa que las crea.

3.1 Uso de variables locales

3.1.1 Estructura de variable local

Disponemos de dos estructuras para crear variables locales

```
<< → v1 v2 v3 << programa que usa v1 v2 v3>> >>
```

⁵Es decir, el espacio de memoria que ocupan queda liberado.

⁶Tiempo que tarda la calculadora en guardar o recuperar el valor que contiene la variable.

y

```
<< → v1 v2 v3 'expresión algebraica que contiene v1 v2 v3' >>
```

En la sintaxis hemos usado, por simplicidad, sólo 3 variables locales: v1,v2 y v3, pero puede usarse cualquier número de variables locales; también puede usarse cualquier nombre válido como nombre de variable local.

Al ejecutarse una estructura de variable local como las descritas arriba, se producen las siguientes acciones:

- Se descargan 3 valores de la pila y se guardan ordenadamente en las variables v1, v2, v3, empezando de derecha a izquierda.
- Se ejecuta el programa o se evalúa la expresión algebraica que acompaña la definición de las variables.

Veamos con mayor detalle como se produce la asignación de variables.

- En primer lugar se descarga el *valor del nivel 1* de la pila, y se guarda en la variable en v3.
- El *valor del nivel 2* de la pila se guarda en v2.
- El *valor del nivel 3* de la pila se guarda en v1.

Este funcionamiento puede parecer algo extraño, pero si escribimos 3 valores en la línea de edición separados por espacios

```
DEG R4Z HEX R= 'X'
{HOME CILIN}
4:
3:
2:
1:
1 2 3
Abas
```

y pulsamos [ENTER], obtenemos:

```
DEG R4Z HEX R= 'X'
{HOME CILIN}
5:
4:
3:
2:
1:
1 2 3
Abas
```

Es decir, si escribimos

```

DEG R42 HEX R= 'X'
{HOME CILIN}
4:
3:
2:
1:
1 2 3
Abas

```

y ejecutamos el programa, entonces:

- El valor 3 se guardaría en v3.
- El valor 2 se guardaría en v2.
- El valor 1 se guardaría en v1.

Es importante destacar que las variables sólo son accesibles por el programa o expresión algebraica que sigue a la declaración de variables locales → v1 v2 v3.

Como primer ejemplo, vamos a escribir un programa que realiza las siguientes acciones:

- Toma del nivel 2 de la pila el valor del radio y lo guarda en la variable local R.
- Toma del nivel 1 de la pila el valor de la altura y lo guarda en la variable local H.
- Calcula el valor del volumen del cilindro usando las variables locales R y H y devuelve el resultado a la pila.

Actividad 3.1 Realiza los siguientes pasos

1. Crea un subdirectorio de HOME con el nombre CIL2, entra en CIL2.

```

DEG R42 HEX R= 'X'
{HOME CIL2}
5:
4:
3:
2:
1:

```

2. Pulsa \uparrow [+] para escribir los delimitadores del programa.

```

DEG R&Z HEX R= 'X'          PRG
{HOME CIL2}
0:
1:
2:
1:
*  ←
*

```

3. Pulsa la tecla \rightarrow [0]



para escribir el carácter que define las variables locales.

```

DEG R&Z HEX R= 'X'          PRG
{HOME CIL2}
0:
1:
2:
1:
* →
*

```

4. Escribe el nombre de las variables locales en el orden R H

```

DEG R&Z HEX R= 'X'          PRG
{HOME CIL2}
0:
1:
2:
1:
* → R H
*

```

en ese orden, el radio R se tomará del nivel 2 de la pila y la altura H del nivel 1.

5. Escribe la expresión algebraica que corresponde al volumen del cilindro entre apóstrofos.

```

DEG R&Z HEX R= 'X'      ALG PRG
{HOME CIL2}
00:
04:
08:
1:
* → R H 'π*R^2*H' ♦
*

```

6. Pulsa [ENTER] para cargar el programa en la pila.

```

DEG R&Z HEX R= 'X'
{HOME CIL2}
00:
04:
08:
1: * → R H 'π*R^2*H' *

```

7. Escribe el nombre Vol

```

DEG R&Z HEX R= 'X'
{HOME CIL2}
00:
04:
08: * → R H 'π*R^2*H' *
1: 'Vol'

```

y pulsa [STO] para guardar el programa con ese nombre.

```

DEG R&Z HEX R= 'X'
{HOME CIL2}
00:
04:
08:
1:
Vol

```

8. El diagrama de pila para el programa Vol es el siguiente:

Programa Vol					
Datos		⇒	Resultado		
Nivel 2	R		Nivel 2		
Nivel 1	H		Nivel 1	Vol	

9. Vamos a usar los siguientes valores de prueba:

R	H	Vol
3.5	12.1	465.6626

Escribe los valores separados por un espacio

```
DEG R↵2 HEX R= 'X'
{HOME CIL2}
4:
0:
0:
0:
1:
3.5 12.1
Vol
```

y pulsa [ENTER].

```
DEG R↵2 HEX R= 'X'
{HOME CIL2}
0:
4:
0:
0:
1:          3.5000
          12.1000
Vol
```

Pulsa [F1] para ejecutar el programa, obtendrás:

```
DEG R↵2 HEX R= 'X'
{HOME CIL2}
0:
4:
0:
0:
1:      π·12.2500·12.1000
Vol
```

Pulsa ↵[ENTER] para ejecutar →NUM, resulta

```
DEG R↵2 HEX R= 'X'
{HOME CIL2}
0:
4:
0:
0:
1:          465.6626
Vol
```

3.1.2 Etiquetado de resultados

Vamos a modificar el programa de la Actividad 3.1 incorporando el comando \rightarrow NUM y añadiendo una etiqueta al resultado.

Actividad 3.2 Realiza los siguientes pasos:

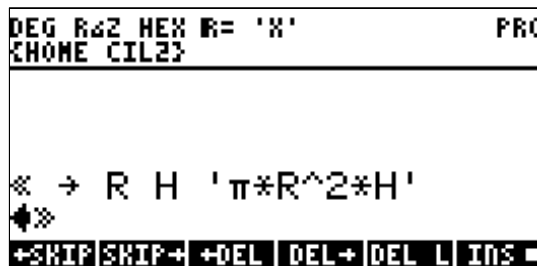
1. Pulsa \uparrow [F1] para recuperar el contenido del programa.

```
DEG R42 HEX R= 'X'
{HOME CIL2}
5:
4:
3:
2:
1:
Vol
```



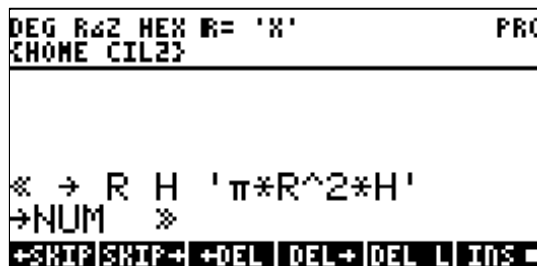
2. Pulsa ∇ para activar el editor.

```
DEG R42 HEX R= 'X' PRG
{HOME CIL2}
* → R H 'π*R^2*H'
*
*SKIP SKIP+ +DEL | DEL+ | DEL L | INS
```



3. Entra el comando \rightarrow NUM.

```
DEG R42 HEX R= 'X' PRG
{HOME CIL2}
* → R H 'π*R^2*H'
→NUM *
*SKIP SKIP+ +DEL | DEL+ | DEL L | INS
```



4. Escribe la etiqueta⁷ "Vol".

⁷La etiqueta es una cadena de caracteres (string) y debe ir entre comillas dobles.

```

DEG R&Z HEX R= 'X' PRG
{HOME CIL2}

« → R H 'π*R^2*H'
→NUM "Vol" »
+SKIP|SKIP+|+DEL|DEL+|DEL L|INS ▀

```

5. Escribe el comando →TAG.

```

DEG R&Z HEX R= 'X' PRG
{HOME CIL2}

« → R H 'π*R^2*H'
→NUM "Vol" →TAG ♦»
OBJ+|+ARRY+|LIST|+STR|+TAG|+UNIT

```

Puedes encontrar el comando en [PRG][TYPES] o en el catálogo de comandos

```

DEG CATALOG: 762 COMMANDS PRG
{HOME
+Qn
+ROW
+STR
+TAG
+TIME
+UNIT
« →
→NL
CANCL OK

```

6. Pulsa [ENTER] para salir del editor y cargar el programa en la pila

```

DEG R&Z HEX R= 'X'
{HOME CIL2}
4:
3:
2:
1: « → R H 'π*R^2*H'
→NUM "Vol" →TAG »
Vol

```

7. Pulsa \uparrow [F1] para actualizar el contenido de la variable Vol con el programa modificado, carga los valores 3.5 y 12.1 en la pila

```

DEG R&Z HEX R= 'X'
{HOME CIL2}
50:
40:
30:
20: 3.5000
1: 12.1000
Vol

```

y pulsa [F1] para ejecutar el programa, obtendrás:

```

DEG R&Z HEX R= 'X'
{HOME CIL2}
50:
40:
30:
20:
1: Vol:465.6626
Vol

```

3.2 Programa con resultados múltiples

Como segundo ejemplo, vamos a realizar un programa con el nombre VACil (Volumen y Area del Cilindro) que lleve a cabo las siguientes acciones:

- Tome del nivel 2 de la pila el valor del radio de la base y lo guarde en la variable local R.
- Tome del nivel 1 de la pila el valor de la altura y lo guarde en la variable local H.
- Calcule, usando las variables locales R y H:
 - El valor del volumen del cilindro.
 - El valor del área lateral.
 - El valor del área total.

También queremos que devuelva los resultados a la pila debidamente etiquetados.

El siguiente diagrama de pila describe con precisión la estructura de las entradas y salidas

Programa VACil			
<i>Datos</i>	⇒	<i>Resultado</i>	
<i>Nivel 3</i>		<i>Nivel 3</i>	Vol: valor volumen
<i>Nivel 2</i>	R	<i>Nivel 2</i>	Alat: valor área lateral
<i>Nivel 1</i>	H	<i>Nivel 1</i>	Atot: valor área total

Usa los siguientes valores iniciales y de prueba:

R	H	Vol	$Alat$	$Atot$
4.5	15.1	960.6	426.9	554.2

Actividad 3.3 Realiza los siguientes pasos:

1. Crea un subdirectorio de HOME con el nombre CIL3.
2. Entra en CIL3.

```
DEG R&Z HEX R= 'X'
{HOME CIL3}
3:
4:
0:
2:
1:
```

3. Escribe el inicio del programa.

```
DEG R&Z HEX R= 'X' PRG
{HOME CIL3}
3:
2:
1:
* → R H*
*
```

4. En este caso, no nos bastará con una sólo expresión algebraica, debemos por lo tanto usar un programa, pulsa $\overline{+}$ para escribir otro par de delimitadores de programa.

```
DEG R&Z HEX R= 'X' PRG
{HOME CIL3}
1:
* → R H
*
*
```

5. Escribe entre apóstrofos la expresión algebraica para el volumen

```

DEG R&Z HEX R= 'X'      ALG PRG
{HOME CIL3}
1:
⊗ → R H
⊗ 'π*R^2*H'
⊗
⊗

```

y escribe el comando \rightarrow NUM para forzar su evaluación⁸ numérica.

```

DEG R&Z HEX R= 'X'      PRG
{HOME CIL3}
1:
⊗ → R H
⊗ 'π*R^2*H' →NUM
⊗
⊗

```

Añade la etiqueta y el comando \rightarrow TAG para generar el valor etiquetado.

```

DEG R&Z HEX R= 'X'      PRG
{HOME CIL3}
⊗ → R H
⊗ 'π*R^2*H' →NUM
"Vol" →TAG ⚡
⊗
⊗

```

Observa que, según el enunciado, el volumen es el resultado que debe aparecer en el nivel más alto de la pila, por lo tanto debe ser cargado en primer lugar. Los siguientes resultados se cargarán en niveles inferiores de la pila y desplazarán el volumen hacia arriba.

6. Procede de forma análoga para calcular el área lateral, evaluarla numéricamente y etiquetarla.

⁸Cuando en un programa aparece una expresión algebraica, la expresión se carga en la pila y debemos usar EVAL o \rightarrow NUM para evaluarla. El caso de la estructura de variable local $\ll \rightarrow R H 'π * R^2 * H' \gg$ es un caso especial. En este caso la expresión algebraica se evalúa automáticamente.

```

DEG R&Z HEX R= 'X' PRG
{HOME CIL3}
« 'π*R^2*H' →NUM
  '2*π*R*H' →NUM
  "A1at" →TAG
  *
  *

```

7. Completa el programa con el cálculo, evaluación y etiquetado del área total,

```

DEG R&Z HEX R= 'X' PRG
{HOME CIL3}
'2*π*R*H' →NUM
"A1at" →TAG
'2*π*R*(H+R)' →NUM
"Atot" →TAG
*
*

```

donde hemos usado la fórmula

$$A_{tot} = A_{lat} + 2A_{bas} = 2\pi RH + 2\pi R^2 = 2\pi R(H + R).$$

8. Pulsa [ENTER] para salir del editor y cargar el programa en la pila.

```

DEG R&Z HEX R= 'X'
{HOME CIL3}
1: « → R H « 'π*R^2*H'
  →NUM "Vol" →TAG '2*
  π*R*H' →NUM "A1at"
  →TAG '2*π*R*(H+R)'
  →NUM "Atot" →TAG »

```

9. Escribe el nombre del programa

```

DEG R&Z HEX R= 'X'
{HOME CIL3}
2: « → R H « 'π*R^2*H'
  →NUM "Vol" →TAG '2*
  π*R*H' →NUM "A1at"
  →TAG '2*π*R*(H+R)'
1: 'VAC11'

```

y pulsa [STO]. Carga los valores de entrada en el orden adecuado

DEG R&Z HEX R= 'X'	
[HOME CIL3]	
5:	
4:	
3:	
2:	4.5000
1:	15.1000
VACIU	

y pulsa [F1] para ejecutar el programa, obtendrás:

DEG R&Z HEX R= 'X'	
[HOME CIL3]	
5:	
4:	
3:	Vol:960.6205
2:	Alat:426.9424
1:	Atot:554.1769
VACIU	

3.3 Uso de variables auxiliares

Al confeccionar algunos programas, es bastante frecuente que necesitemos una *variable auxiliar*. En esta sección vamos a ver como definir y usar *variables locales auxiliares*.

Como ejemplo, vamos a escribir un programa, que denominaremos Med, que tome una lista de números de la pila, la almacene en una variable local X, calcule la media aritmética y devuelva el resultado a la pila con la etiqueta Med.

Aunque es posible realizar el programa utilizando comandos de manipulación de la pila, podemos escribir un programa más claro usando las variables locales N y S en las que guardaremos, respectivamente, el número de elementos de la lista y la suma. El diagrama de pila del programa Med es el siguiente:

Programa Med			
Datos		⇒	Resultado
Nivel 1	Lista X		Nivel 1 Valor medio de los elementos de la lista X

Usa los datos de prueba

X	Med
{1, 2, 3, 4, 5, 6}	3.5

Actividad 3.4 Realiza los siguientes pasos:

1. Crea un subdirectorio de HOME con el nombre MED1.
2. Entra en el subdirectorio MED1

```

DEG R↓Z HEX R= 'X'
CHOME MED1
120040
1:

```

3. Escribe los delimitadores de programa y el siguiente inicio de estructura de variable local.

```

DEG R↓Z HEX R= 'X' PRG
CHOME MED1
⊗ 0 0 → X N S
⊗
⊗
⊗
◀LIST|ELIST|MLIST|SORT|REVLI|ADD

```

La sintaxis $0\ 0 \rightarrow X\ N\ S$, asigna el valor inicial 0 a las variables locales N y S, por lo tanto tomará un sólo objeto de la pila que guardará en la variable X.

4. Escribe el siguiente fragmento de programa

```

DEG R↓Z HEX R= 'X' PRG
CHOME MED1
⊗ 0 0 → X N S
⊗ X ΣLIST 'S' STO ◀
⊗
⊗
◀LIST|ELIST|MLIST|SORT|REVLI|ADD

```

que calcula el valor de la suma de los elementos de la lista y guarda este valor en la variable local S.

5. Usa ahora el comando SIZE para calcular el número de elementos de la lista y guarda el número de elementos en la variable local N.

```

DEG R42 HEX R= 'X' PRG
{HOME MED1}
« 0 0 → X N S
« X ΣLIST 'S' STO
X SIZE 'N' STO
»
»
GET GETI PUT PUTI SIZE POS

```

6. Calcula la división y etiqueta el resultado.

```

DEG R42 HEX R= 'X' PRG
{HOME MED1}
« 0 0 → X N S
« X ΣLIST 'S' STO
X SIZE 'N' STU
'S/N'→NUM "Med" →TAG
»
»
GET GETI PUT PUTI SIZE POS

```

7. Pulsa [ENTER] para salir del editor y cargar el programa en la pila.

```

DEG R42 HEX R= 'X'
{HOME MED1}
2:
1: « 0 0 → X N S « X
ΣLIST 'S' STO X
SIZE 'N' STO 'S/N'
→NUM "Med" →TAG » »

```

Entra el nombre

```

DEG R42 HEX R= 'X'
{HOME MED1}
2: « 0 0 → X N S « X
ΣLIST 'S' STO X
SIZE 'N' STO 'S/N'
→NUM "Med" →TAG » »
1: 'Med'

```

y pulsa [STO] para guardar el programa.

8. Carga la lista de prueba en la pila

```

DEG R&Z HEX R= 'X'
[HOME MED1]
50:
40:
30:
20:
1: (1 2 3 4 5 6)
Med

```

y ejecuta el programa, debes obtener

```

DEG R&Z HEX R= 'X'
[HOME MED1]
50:
40:
30:
20:
1: Med: 3.5000
Med

```

4 Presentación de resultados con MSGBOX

Una buena forma de presentar resultados consiste en añadir una etiqueta al valor usando el comando \rightarrow TAG. Una segunda forma, muy elegante, de presentar mensajes y resultados es el comando MSGBOX. El comando MSGBOX toma como argumento un string y, como resultado, produce un cuadro de diálogo en la pantalla. Puedes encontrar el comando MSGBOX en el [PRG][OUT], o bien en el catálogo de funciones y comandos

```

DEG [CATALOG: 762 COMMANDS]
[CHOM]
40:
30:
20:
1:
[MODULAR]
[MR00T]
[MSGBOX]
[MSLV]
[MSOLVR]
[MULTMOD]
[CANCL] [OK]

```

Un sencillo ejemplo, nos permitirá comprender su funcionamiento.

Actividad 4.1 Realiza los siguientes pasos.

1. Escribe el siguiente programa:

```

DEG R&Z HEX R= 'X'
{HOME MED1}
3:
2:
1:  «
    "TEXTO DE EJEMPLO"
    MSGBOX »
PVIEW TEXT |CLLCD DISP |FREEZ|MSGBO

```

2. Pulsa [EVAL] para ejecutar el programa, obtendrás:

```

DEG R&Z HEX R= 'X'
{HOME MED1}
3:
2:
1:
    [TEXT BOX]
    "TEXTO DE EJEMPLO"
    MSGBOX »
                                OK

```

3. Pulsa [ENTER] o [F6] para ejecutar [OK] y continuar.

El argumento de MSGBOX debe ser una cadena de caracteres (string). El siguiente ejemplo muestra como crear un mensaje a partir de un valor usando el comando \rightarrow STR y la concatenación de strings. El objetivo es escribir un programa denominado AR1 que

- Tome de la pila el valor del radio y lo almacene en la variable local R.
- Calcule el valor del área del círculo de radio R.
- Presente un mensaje con el texto: AREA = valor calculado.
- Cargue en la pila el resultado convenientemente etiquetado.

Usa como valores de prueba

R	AREA
23.41	1721.68

Actividad 4.2 Realiza los siguientes pasos:

1. Crea un subdirectorio de HOME con el nombre DRP5.
2. Entra en DRP5

```

DEG R42 HEX R= 'X'
<HOME DRP5>
5:
4:
3:
2:
1:

```

3. Escribe los delimitadores de programa y el programa

```

DEG R42 HEX R= 'X'
<HOME DRP5>
4:
3:
2:
1:  « → R « 'π*R^2'
      →NUM » »
AR1

```

que calcula el área a partir del radio. Guarda el programa con el nombre AR1

```

DEG R42 HEX R= 'X'
<HOME DRP5>
5:
4:
3:
2:
1:

```

↓

```

AR1 28.4100

```

y usa los valores de prueba para verificar su funcionamiento.

```

DEG R42 HEX R= 'X'
<HOME DRP5>
5:
4:
3:
2:
1:

```

```

AR1 1,721.6811

```

4. Pulsa \uparrow [F1] para recuperar el programa y \downarrow para activar el editor.

```
DEG R&Z HEX R= 'X'
{HOME DRP5}
4:
3:
2:
1: « → R « 'π*R^2'
    →NUM » »
ARR1
```

Necesitaremos dos copias del resultado, una para construir el mensaje con MSGBOX y otra para devolverla etiquetada a la pila. Podríamos escribir el programa con comandos de pila pero, para mayor claridad, usaremos la variable local auxiliar A que contendrá el resultado. Modifica el programa como sigue:

```
DEG R&Z HEX R= 'X' PRG
{HOME DRP5}
« 0 → R A
  « 'π*R^2' →NUM
  'A' STO
  »
»
↳SKIP ↳SKIP+ ↳DEL ↳DEL+ ↳DEL L ↳INS ▢
```

Ahora el programa toma un valor de la pila y lo guarda en la variable local R, la variable A contendrá inicialmente el valor 0. Una vez calculado el valor del área, lo guarda en A.

5. Escribe la siguiente línea de programa.

```
DEG R&Z HEX R= 'X' PRG
{HOME DRP5}
« 0 → R A
  « 'π*R^2' →NUM
  'A' STO
  "AREA = " A →STR +
  »
↳OBJ+ ↳ARRY ↳LIST ↳STR ↳TAG ↳UNIT
```

Esta línea de código produce el siguiente efecto:

- Primero carga la cadena de caracteres "Area = " en la pila.
- Carga en la pila el valor de A y lo convierte en string con el comando →STR.
- Concatena⁹ las dos cadenas de caracteres con el comando +.

⁹El comando +, cuando actúa con strings, realiza la concatenación, esto es, junta las dos cadenas.

Para comprender bien el efecto de la nueva línea de código, pulsa [ENTER] para salir del editor y cargar el programa en la pila,

```
DEG R42 HEX R= 'X'
{HOME DRP5}
2:
1: « 0 → R A « 'π*R^2'
   →NUM 'A' STO
   "AREA = " A →STR +
   » »
AR1
```

guarda la nueva versión del programa con \uparrow [F1], carga el valor de prueba en la pila

```
DEG R42 HEX R= 'X'
{HOME DRP5}
0:
4:
0:
2:
1: 23.4100
AR1
```

y ejecuta el programa, obtendrás el siguiente string:

```
DEG R42 HEX R= 'X'
{HOME DRP5}
0:
4:
0:
2:
1: "AREA = 1721.6811"
AR1
```

que constituye el texto de entrada para el comando MSGBOX.

6. Borra la pila, recupera nuevamente el programa y activa el editor, añade el comando MSGBOX.

```
DEG R42 HEX R= 'X' PRG
{HOME DRP5}
« 0 → R A
  « 'π*R^2' →NUM 'A'
STO "AREA = " A →STR
+ MSGBOX ↓
»
←SKIP SKIP← +DEL | DEL← | DEL L | INS ▀
```

7. Para terminar, escribe la última línea de código

```
DEG R&2 HEX R= 'X' PRG
{HOME DRP5}
STO "AREA = " A →STR
MSGBOX
A "AREA" →TAG
```

que carga el valor del área en la pila, carga la etiqueta y construye el valor etiquetado con →TAG.

8. Pulsa [ENTER] para salir del editor y cargar el programa en la pila,

```
DEG R&2 HEX R= 'X'
{HOME DRP5}
1: « 0 → R A « 'π*R^2'
→NUM 'A' STO
"AREA = " A →STR +
MSGBOX A "AREA"
→TAG » »
```

pulsa \uparrow [F1] para guarda la versión actual en AR1 y carga el valor de prueba en la pila.

```
DEG R&2 HEX R= 'X'
{HOME DRP5}
5:
4:
3:
2:
1: 23.4100
```

Cuando ejecutes el programa, obtendrás en primer lugar el siguiente mensaje:

```
DEG R&2 HEX R= 'X'
{HOME DRP5}
5: AREA =
4: 1721.6811
3:
2:
1: 23.4100
OK
```


Actividad 5.1 Crea un directorio con el nombre DRP6, fija el modo angular en radianes y entra en DRP6.

```
RAD R2 HEX R= 'X'
CHOME DRP6
1:
2:
3:
4:
5:
6:
7:
8:
9:
10:
```

Realiza los siguientes pasos:

1. Escribe los delimitadores de programa e inicia una estructura de variable local.

```
RAD R2 HEX R= 'X' PRG
CHOME DRP6
1:
2: X → X
3: X ↕
4: X
5: X
6: X
```

2. El submenú¹⁰ [BRCH] del menú [PRG]

```
RAD R2 HEX R= 'X' PRG
CHOME DRP6
1:
2: X → X
3: X ↕
4: X
5: X
6: X
STACK MEN BRCH TEST TYPE LIST
```

contiene los órdenes de ejecución condicionada, accede a [BRCH][IF] y escribe la condición.

```
RAD R2 HEX R= 'X' ALG PRG
CHOME DRP6
1:
2: X → X
3: IF 'X<0' ↕
4: X
5: X
6: X
IF THEN ELSE END BRCH
```

¹⁰BRanCH = ramificar.

Observa que hemos usado notación algebraica para la condición y, por lo tanto, debemos delimitarla con apóstrofes.

3. Pulsa¹¹ $\uparrow[\cdot]$ para escribir un retorno de carro. Escribe la palabra clave THEN y, a continuación, las acciones que deben ejecutarse si la condición es cierta.

```

RAD R42 HEX R= 'X'          ALG PRG
{HOME DRP6}
* → X
* IF 'X<0'
THEN 'SIN(X)'
*
*
IF THEN ELSE END BRCH

```

4. Entra un retorno de carro, escribe la palabra ELSE y, a continuación, las acciones que deben ejecutarse cuando la condición es falsa.

```

RAD R42 HEX R= 'X'          ALG PRG
{HOME DRP6}
* → X
* IF 'X<0'
THEN 'SIN(X)'
ELSE 'X^2'
*
*
IF THEN ELSE END BRCH

```

5. Escribe la palabra clave END para cerrar la estructura de ejecución condicional.

```

RAD R42 HEX R= 'X'          PRG
{HOME DRP6}
* IF 'X<0'
THEN 'SIN(X)'
ELSE 'X^2' END
*
*
IF THEN ELSE END BRCH

```

6. En las estructuras condicionales, las condiciones (que siguen a la palabra clave IF) se evalúan automáticamente; por el contrario, las expresiones algebraicas que aparecen después de THEN o de ELSE sólo se cargan en la pila, por lo tanto, en cualquier caso deben ser evaluadas. Por lo tanto, debes incluir un EVAL o un \rightarrow NUM después de la palabra clave END.

¹¹Tecla $\uparrow(10,3)$.

```

RAD R42 HEX R= 'X'          PRG
{HOME DRP6}
« IF 'X<0'
THEN 'SIN(X)'
ELSE 'X^2' END →NUM
»
»
»
IF | THEN | ELSE | END | | BRCH

```

Pulsa [ENTER] para cargar el programa en la pila.

```

RAD R42 HEX R= 'X'          PRG
{HOME DRP6}
0:
2:
1: « → X « IF 'X<0'
   THEN 'SIN(X)' ELSE
   'X^2' END →NUM » »
IF | THEN | ELSE | END | | BRCH

```

7. Observa que los retornos de carro desaparecen al cargar el programa en la pila, pero si pulsas [▼] para acceder al editor, obtendrás la siguiente presentación.

```

RAD R42 HEX R= 'X'          PRG
{HOME DRP6}
▼ → X
  «
    IF 'X<0'
    THEN 'SIN(X)'
    ELSE 'X^2'
+SKIP|SKIP+ | +DEL | DEL+ | DEL L | INS ▣

```

Pulsa [ENTER] para regresar a la pila y guarda el programa con el nombre F. Pulsa [VAR], carga el valor 4 en la pila y pulsa [F1] para ejecutar F,

```

RAD R42 HEX R= 'X'          PRG
{HOME DRP6}
F:
4:
0:
2:
1:
F | | | | | 4

```

obtendrás:

```

RAD R&Z HEX R= 'X'
{HOME DRP6}
5:
4:
3:
2:
1: 16.0000
F

```

Entra ahora el valor -3 y ejecuta F, ahora el resultado es

```

RAD R&Z HEX R= 'X'
{HOME DRP6}
5:
4:
3:
2: 16.0000
1: -0.1411
F

```

Verifica que este valor es el correspondiente a seno de -3 radianes.

Actividad 5.2 Define y representa la siguiente función

$$g(x) = \begin{cases} \cos x & \text{si } x \leq 0, \\ 1 - x^2 & \text{si } x > 0. \end{cases}$$

Verifica que funciona correctamente con los valores de prueba $x = -0.25$ rad y $x = 1.25$.

Actividad 5.3 Modifica la función anterior para que fije el modo angular en radianes antes de realizar la evaluación.

5.2 Estructura CASE

La estructura CASE permite gestionar más de dos alternativas, la sintaxis de la estructura CASE es

```

<< ...
CASE
'condición 1' THEN 'acciones para condición 1 cierta' END
'condición 2' THEN 'acciones para condición 2 cierta' END
      ⋮
'condición n' THEN 'acciones para condición n cierta' END
'acciones por defecto'
END ...
>>

```



```

RAD R42 HEX R= 'X'          PRG
[HOME DRP6]
* → X
* CASE
'X ≤ -π' THEN '-1' END
*
*
CASE | THEN | END | | | BRCH

```

4. Para escribir la condición 2, dado que la evaluación de condiciones es secuencial empezando por la primera basta con que exijas $x \leq 0$.

```

RAD R42 HEX R= 'X'          PRG
[HOME DRP6]
* → X
* CASE
'X ≤ -π' THEN '-1' END
'X ≤ 0' THEN 'COS(X)' END
*
*
CASE | THEN | END | | | BRCH

```

5. Escribe la línea correspondiente a la condición 3.

```

RAD R42 HEX R= 'X'          PRG
[HOME DRP6]
* → X
* CASE
'X ≤ -π' THEN '-1' END
'X ≤ 0' THEN 'COS(X)' END
'X ≤ 2' THEN 1 END
*
*
CASE | THEN | END | | | BRCH

```

6. Si no se ha cumplido ninguna de las condiciones anteriores, esto es, para $x > 2$, la función debe asignar el valor $5 - x^2$, incluimos la fórmula como acción por omisión y cerramos la estructura CASE con un END final.

```

RAD R42 HEX R= 'X'          PRG
[HOME DRP6]
* CASE
'X ≤ -π' THEN '-1' END
'X ≤ 0' THEN 'COS(X)' END
'X ≤ 2' THEN 1 END
'5 - X^2' END
*
CASE | THEN | END | | | BRCH

```

7. De forma análoga a como hicimos en la actividad anterior, escribimos el comando →NUM

```

RAD R&Z HEX R= 'X'          PRG
{HOME DRP}
'X<=-π' THEN '-1' END
'X<=0' THEN 'COS(X)' END
'X<=2' THEN '1' END
'5-X^2' END →NUM ♦
♦
CASE THEN END | | | BRCH

```

que se ejecutará después de salir de la estructura CASE, provocando la evaluación de la expresión algebraica cargada en la pila. Pulsa [ENTER] para cargar el programa en la pila

```

RAD R&Z HEX R= 'X'          PRG
{HOME DRP}
1: ♦ → X ♦ CASE 'X<=-π'
  THEN '-1' END 'X<=0'
  THEN 'COS(X)' END '
  X<=2' THEN '1' END '5-
  X^2' END →NUM ♦ ♦
CASE THEN END | | | BRCH

```

y guárdalo con el nombre G.

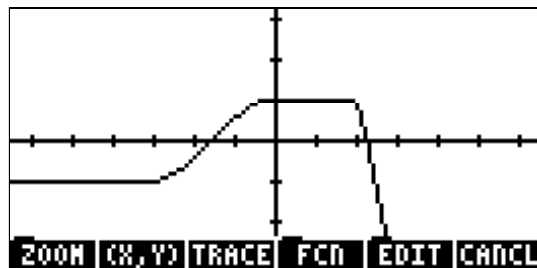
8. Puedes verificar con diferentes valores el buen funcionamiento de G, si entras en la aplicación de representación de funciones y escribes

```

PLOT - FUNCTION
Y(X)=G(X)
EDIT | ADD | DEL | CHOOSERASE | DRAW

```

obtendrás la siguiente representación gráfica:



Actividad 5.5 *Escribe un programa que permita calcular la siguiente función*

$$g(x) = \begin{cases} -2 & \text{si } x \leq -1, \\ x - 1 & \text{si } -1 < x \leq 0, \\ x^2 - 1 & \text{si } 0 < x \leq 2, \\ 1 & \text{si } 2 < x. \end{cases}$$

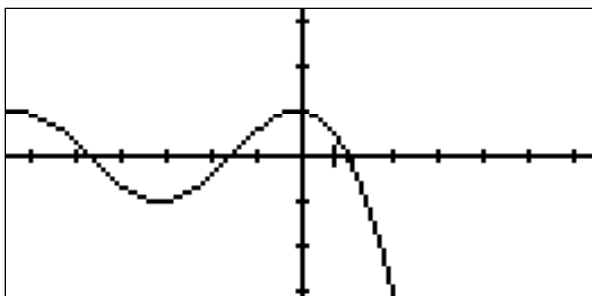
Actividad 5.6 *Realiza manualmente un esquema del gráfico de la función g definida en la actividad anterior. Representa la función con la calculadora y compara el resultado con tu gráfico.*

5.3 Soluciones de las actividades

Actividad 5.2

Programa	Acción
<<	Abre programa.
→ X	Define X como variable local.
<<	Inicia programa asociado a variable local.
IF 'X ≤ 0'	Inicia estructura condicional IF.
THEN 'cos(x)'	Expresión del caso afirmativo.
ELSE '1-x^2'	Expresión en el caso negativo.
END	Final estructura condicional.
→NUM	Evalúa la expresión cargada por IF.
>>	Cierra estructura de variable local.
>>	Cierra programa.

El gráfico de la función es el siguiente:



Actividad 5.3

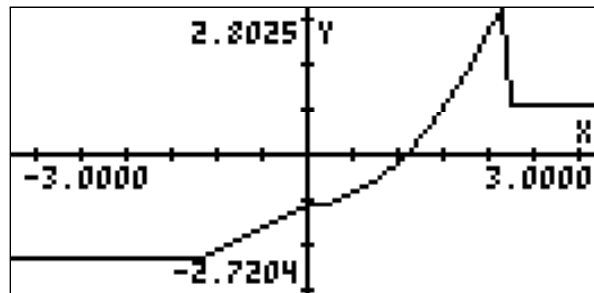
Simplemente debes escribir RAD delante de →NUM.

Actividad 5.5

Programa	Acción
<<	Abre programa.
→ X	Define X como variable local.
<<	Inicia programa asociado a variable local.
CASE	Inicia estructura condicional CASE
'X ≤ -1' THEN '-2' END	Primer caso.
'X ≤ 0' THEN 'X-1' END	Segundo caso.
'X ≤ 2' THEN 'X^2-1' END	Tercer caso.
1	Acción por omisión.
END	Final estructura condicional CASE.
→NUM	Evalúa la expresión cargada por CASE.
>>	Cierra estructura de variable local.
>>	Cierra programa.

Actividad 5.6

La representación gráfica en el intervalo $x \in [-3, 3]$ tiene el siguiente aspecto:



Observa la discontinuidad de salto en $x = 2$.

6 Formularios de entrada de datos

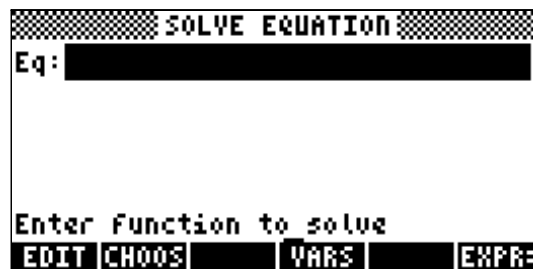
6.1 Formularios de entrada de datos

Cuando los datos son simples, la mejor estrategia de entrada de datos para nuestros programas consiste en que el programa tome los datos directamente de la pila. En este caso basta con escribir un simple diagrama de pila para describir la forma de funcionamiento del programa. Cuando los datos de entrada son complejos, los formularios de entrada de datos son un excelente recurso.

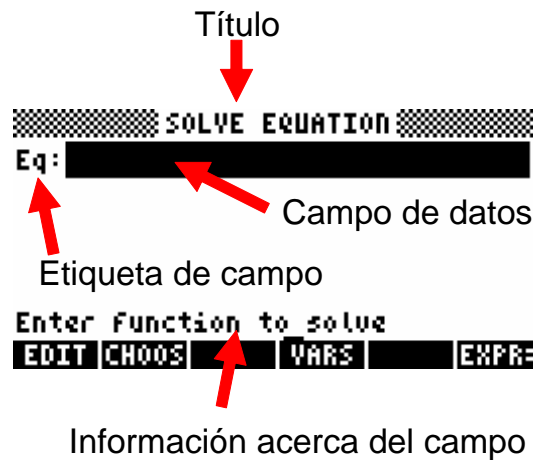
Pulsa \uparrow [7] para acceder al menú [N.SLV] que contiene los recursos de resolución numérica de ecuaciones.



Pulsa [ENTER] para aceptar la opción Solve equation, obtendrás el siguiente formulario.



Observa los siguientes elementos:



Un formulario contiene:

- Un título principal, en el ejemplo es SOLVE EQUATION.
- Uno o más campos de entrada de datos.
- Delante de cada campo, hay una etiqueta.
- Cuando seleccionamos un campo, puede mostrarse información de ayuda en la base de la pantalla, en el ejemplo vemos la información *Enter function to solve*.

6.2 Uso simplificado del comando INFORM

El comando INFORM nos permite definir formularios de entrada de datos para nuestros programas. Puedes acceder al comando INFORM en [PRG][IN] o a través del catálogo de funciones.



El comando INFORM toma como entrada 5 objetos:

- Nivel 5: “*Título del formulario*”, debe ser un *string*.
- Nivel 4: {*campo1, campo2, ...*}, lista de especificaciones de campos.
- Nivel 3: {*formato*}, lista de especificaciones de formato, puede ser una lista vacía.
- Nivel 2: {*lista de valores de reset*}, puede ser una lista vacía.
- Nivel 1: {*lista de valores iniciales*}, puede ser vacía.

A partir de la información proporcionada por estos 5 objetos, la calculadora construye el formulario de entrada de datos y suspende la ejecución del programa hasta que pulsemos [OK] o [CANCEL].

Si el usuario llena el formulario y pulsa [OK], se obtiene:

- Nivel 2: Una lista de los valores entrados en los distintos campos del formulario.
- Nivel 1: El valor 1, que indica el usuario ha salido del formulario pulsando [OK].

Si el usuario cancela el formulario, se obtiene:

- Nivel 1: El valor 0, que indica que el usuario no ha rellenado el formulario.

Aunque en principio puede parecer complejo, en la práctica es bastante sencillo usar el comando INFORM. Como primer ejemplo vamos a construir un formulario con el título PRUEBA y tres campos de entrada con etiquetas A, B y C.

Actividad 6.1 Crea un subdirectorio de HOME con el nombre DRP7, entra en DRP7 y realiza los siguientes pasos:

1. Escribe los delimitadores de programa y un string con el título del formulario.

```

RAD R42 HEX R= 'X'          PRG
{HOME DRP7}
2:
1:
* "PRUEBA"
*

```

2. Pulsa¹² $\uparrow[\cdot]$ para entrar un retorno de carro y escribe la lista con las etiquetas de los campos.

```

RAD R42 HEX R= 'X'          PRG
{HOME DRP7}
2:
1:
* "PRUEBA"
{"A" "B" "C"}
*

```

Observa que las etiquetas son strings y que, por lo tanto, deben ir entre comillas dobles.

3. Entra tres listas vacías, correspondientes a las especificaciones de formato, valores de reset y valores por defecto.

```

RAD R42 HEX R= 'X'          PRG
{HOME DRP7}
1:
* "PRUEBA"
{"A" "B" "C"}
{}{}{}
*

```

Eso completa el grupo de 5 objetos necesarios para ejecutar INFORM.

4. Accede a [PRG][IN] y entra el comando INFORM.

¹²Tecla $\uparrow(10,3)$.

```

RAD R4Z HEX R= 'X'          PRG
[HOME DRP?]
1:
* "PRUEBA"
("A" "B" "C")
()()() INFORM *
*
INFOR|NOVAL|CHOOS|INPUT|KEY|WAIT

```

5. Pulsa [ENTER] para cargar el programa en la pila y guárdalo con el nombre P1.

```

RAD R4Z HEX R= 'X'          PRG
[HOME DRP?]
3:
2: * "PRUEBA" ( "A"
   "B" "C" ) ( ) ( ) (
   ) INFORM *
1: 'P1'

```

6. Accede al área de variables y pulsa [F1] para ejecutar el programa.

```

RAD R4Z HEX R= 'X'          PRG
[HOME DRP?]
5:
4:
3:
2:
1:
F1

```

Como resultado, la calculadora construirá el formulario de entrada que acabas de diseñar

```

PRUEBA:
A
B
C
EDIT | | | CANCL | OK

```

Observa el título, las etiquetas de campo y las opciones [CANCL] y [OK].

7. Entra los valores $A = 1$ y $B = 3$, deja en blanco el campo C .

8. Para salir del formulario, pulsa [ENTER] o [F6] para ejecutar [OK], obtendrás

El valor 1 del Nivel 1 de la pila, indica que el usuario ha rellenado el formulario; la lista del Nivel 2 contiene los valores de los campos. Observa que el valor correspondiente al campo C es NOVAL, esto es debido a que hemos dejado el campo vacío.

Actividad 6.2 Ejecuta el programa P1 y, cuando aparezca el formulario, pulsa [F5] para ejecutar [CANCEL]. Observa que, como resultado, obtienes el valor 0 en el Nivel 1 de la pila. Esto indica que el usuario ha cancelado el formulario.

Actividad 6.3 Vamos a entrar valores por omisión al formulario del programa P1. Realiza los siguientes pasos:

1. Pulsa ⌈[F1] para recuperar el contenido de P1 y [▼] para activar el editor. Según se ha indicado anteriormente, la tercera lista vacía

sirve para especificar los valores iniciales.

2. Escribe tres ceros en esa lista.

```

RAD R42 HEX R= 'X'          PRG
CHOME DRP??
⊗ "PRUEBA" ( "A" "B"
"C" ) ( ) ( )
(0 0 0 )*
INFORM
⊗
+SKIP|SKIP+|+DEL|DEL+|DEL L|INS

```

Esto indica que los tres campos deben tomar, inicialmente, el valor 0.

3. Pulsa [ENTER] para salir del editor y cargar el programa en la pila y \uparrow [P1] para guardar en P1 la nueva versión. A continuación, ejecuta P1, obtendrás

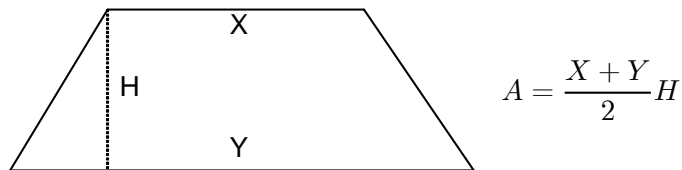
```

PRUEBA
A 0
B 0
C 0
EDIT | | | CANCEL | OK

```

6.3 Un ejemplo de aplicación

Como ejemplo, vamos a escribir un programa que calcule el área de un trapecio



y que tenga como entrada un formulario. Podemos realizar el programa de varias maneras, optaremos por una solución mixta, esto es, crearemos una función con el nombre A que tome de la pila los valores X, Y y H y que devuelva el valor del área debidamente etiquetada. Después crearemos un programa que genere un formulario y que ejecute la función A, de esta forma puedes calcular el área de forma directa y a través del formulario.

Actividad 6.4 Realiza los siguientes pasos:

1. En primer lugar, crea un subdirectorio de HOME con el nombre DRP8, y entra en él.

```

RAD R&Z HEX R= 'X'
{HOME DRP}
0:
4:
0:
0:
0:
1:

```

2. Escribe la función que calcula el área en el editor de ecuaciones.

```

A(X,Y,H)= $\frac{X+Y}{2}$ ·H

```

↑

```

EDIT CURS | BIG = | EVAL | FACTO | SIMP

```

Recuerda que para que aparezcan las comas dentro del paréntesis debes entrar un espacio¹³.

3. Pulsa [ENTER] para cargar la fórmula en la pila

```

RAD R&Z HEX R= 'X'
{HOME DRP}
4:
0:
0:
0:
1:

```

$A(X,Y,H)=\frac{X+Y}{2} \cdot H$

y pulsa \uparrow [2] para ejecutar el comando DEFINE, obtendrás:

```

RAD R&Z HEX R= 'X'
{HOME DRP}
0:
4:
0:
0:
0:
1:

```

A

4. Pulsa \uparrow [F1] para recuperar el contenido de A,

¹³Tecla (10,4).

```

RAD R42 HEX R= 'X'
[HOME DRP8]
4:
3:
2:
1: « → X Y H '(X+Y)/2*
    H' »

```

pulsa [▼] para activar el editor.

```

RAD R42 HEX R= 'X'
[HOME DRP8]
« → X Y H '(X+Y)/2*H'
»

```

Añade el comando \rightarrow NUM, para forzar la evaluación numérica, el string "AREA" y el comando¹⁴ \rightarrow TAG para generar la etiqueta del resultado.

```

RAD R42 HEX R= 'X'
[HOME DRP8]
« → X Y H '(X+Y)/2*H'
→NUM "AREA" →TAG »

```

5. Pulsa [ENTER] para salir del editor y cargar el programa en la pila.

```

RAD R42 HEX R= 'X'
[HOME DRP8]
3:
2:
1: « → X Y H '(X+Y)/2*
    H' →NUM "AREA" →TAG
    »

```

¹⁴Recuerda que puedes obtener \rightarrow TAG en [PRG][TYPE].

Pulsa [CANCEL] y borra la pila.

- Ahora nos queda por realizar la parte nueva, que consiste en tomar los valores que nos devuelve INFORM, cargarlos en la pila y ejecutar A para calcular el área. Pulsa ↵[ATRAP] para cargar el programa en la pila y [▼] para activar el editor

```

RAD R4Z HEX R= 'X'          PRG
[HOME DRP8]
« "AREA TRAPECIO" (
  "X" "Y" "H" } { } { }
  { } INFORM
  ♦»
+SKIP|SKIP+|+DEL|DEL+|DEL L|INS ▣

```

- En esta primera versión no tendremos en cuenta si el usuario ha cancelado el formulario de entrada, escribe el comando¹⁵ DROP

```

RAD R4Z HEX R= 'X'          PRG
[HOME DRP8]
« "AREA TRAPECIO" (
  "X" "Y" "H" } { } { }
  { } INFORM
  DROP ♦»
DUP|SWAP|DROP|OVER|ROT|UNROT

```

para borrar el valor 0 ó 1 del nivel 1 que resulta después de ejecutarse INFORM.

- A continuación escribe el comando EVAL

```

RAD R4Z HEX R= 'X'          PRG
[HOME DRP8]
« "AREA TRAPECIO" (
  "X" "Y" "H" } { } { }
  { } INFORM
  DROP EVAL ♦»
DUP|SWAP|DROP|OVER|ROT|UNROT

```

que romperá la lista de valores y cargará su contenido en la pila.

- Para acabar, simplemente escribe A.

¹⁵Puedes obtener DROP en [PRG][STACK], o en el catálogo de comandos. También puedes teclearlo directamente.

```

RAD R&Z HEX R= 'X'          PRG
[HOME DRP]
« "AREA TRAPECIO" (
  "X" "Y" "H" ) { } { }
{ } INFORM
DROP EVAL A»
  DUP | SWAP | DROP | OVER | ROT | UNROT

```

Como existe un programa con el nombre A en el directorio activo, esto provocará la ejecución del programa. No hace falta nada más.

13. Pulsa [ENTER] para salir del editor y cargar el programa en la pila,

```

RAD R&Z HEX R= 'X'          PRG
[HOME DRP]
2:
1: « "AREA TRAPECIO" (
  "X" "Y" "H" ) { } { }
  { } INFORM DROP
  EVAL A »
  ATRAP | A | | | |

```

pulsa ↵[ATRAP] para guardar la nueva versión del programa.

14. Vamos a verificar el funcionamiento de ATRAP, pulsa [F1] para ejecutar el programa, obtendrás el formulario.

AREA TRAPECIO	
X	
Y	
H	
EDIT CANCL OK	

Entra los valores de prueba

AREA TRAPECIO	
X	12.1
Y	4.7
H	5.8
EDIT CANCL OK	



y pulsa [OK] o [ENTER], obtendrás:

```

RAD R4Z HEX R= 'X'
[HOME DRP2]
5:
4:
3:
2:
1:
AREA: 48.7200
ATRAP A

```

Actividad 6.5 La primera versión del programa ATRAP no ha previsto el caso de que el usuario cancele el formulario de entrada, en este caso en el Nivel 2 de la pila no habrá lista de valores y se producirá un error. Vamos a verificar esta afirmación, ejecuta¹⁶ [CLEAR] para limpiar la pila

```

RAD R4Z HEX R= 'X'
[HOME DRP2]
5:
4:
3:
2:
1:
ATRAP A

```

y pulsa [F1] para ejecutar ATRAP, obtendrás nuevamente el formulario de entrada.

```

AREA TRAPECIO
X
Y
H
EDIT CANCL OK

```



Cancela el formulario pulsando [CANCEL] o [F5], obtendrás el siguiente error:

```

RAD R4Z HEX R= 'X'
[HOME DRP2]
5:
4:
3:
2:
1:
ATRAP A

```

⚠ EVAL Error:
Too Few
Arguments

¹⁶Tecla ⌈(4,5).

El error se ha producido como sigue: al cancelar INFORM, se ha devuelto únicamente un valor 0 que se ha cargado en el nivel 1, el comando DROP ha borrado el valor 0; entonces, el comando EVAL ha encontrado la pila vacía y se ha generado el error.

Vamos a modificar el programa *ATRAP* para que funcione correctamente en el caso de que el usuario cancele el formulario de entrada de datos. Para ello vamos a emplear la estructura de ejecución condicionada IF-THEN-ELSE. La secuencia de acciones es la siguiente:

1. Tomamos el valor 0 o 1 que resulta de la ejecución de INFORM y lo guardamos en la variable local F (flag).
2. Si F toma el valor 1, tendremos una lista de valores en el Nivel 2 de la pila y debemos ejecutar A para calcular el área.
3. Si F toma el valor 0, el usuario ha cancelado el formulario y no tenemos lista de datos, podemos optar por no hacer nada o bien escribir un mensaje, por ejemplo, CANCELADO.

Actividad 6.6 Realiza los siguientes pasos.

1. Recupera el contenido de *ATRAP* y activa el editor.

```

RAD R42 HEX R= 'X'          PRG
{HOME DRP}
* "AREA TRAPECIO" (
"X" "Y" "H" } { } { }
{ } INFORM DROP EVAL
A
*

```

2. Borra los tres últimos comandos del programa.

```

RAD R42 HEX R= 'X'          PRG
{HOME DRP}
* "AREA TRAPECIO" (
"X" "Y" "H" } { } { }
{ } INFORM
*

```

3. Crea la estructura de variable local

```

RAD R42 HEX R= 'X'          PRG
{HOME DRP}
* "AREA TRAPECIO" (
"X" "Y" "H" } { } { }
{ } INFORM
→ F
*

```

seguida de un nuevo par de delimitadores de programa.

```

RAD R42 HEX R= 'X'          PRG
{HOME DRP}
"X" "Y" "H" } { } { }
{ } INFORM
→ F
*

```

Observa que una vez ejecutado `INFORM`, el valor 0 ó 1 del Nivel 1 de la pila se guardará en la variable local `F`.

4. Accede a `[PRG][BRCH][IF]` y escribe la condición.

```

RAD R42 HEX R= 'X'          PRG
{HOME DRP}
"X" "Y" "H" } { } { }
{ } INFORM
→ F
* IF F

```

Si el contenido de `F` es 1, se ejecutará el caso cierto, si `F` vale 0, se ejecutará el caso falso.

5. Escribe las acciones para el caso cierto.

```

RAD R42 HEX R= 'X'          PRG
{HOME DRP}
"X" "Y" "H" } { } { }
{ } INFORM
→ F
* IF F
THEN EVAL A

```

Observa que ahora no necesitas el comando DROP, pues el valor del Nivel 1 de la pila se ha descargado de la pila al guardarse en la variable local F.

6. Escribe las acciones para el caso falso, que consistirán en construir un cuadro de mensaje¹⁷ con el texto CANCELADO.

```

RAD R42 HEX R= 'X'                                PRG
{HOME DRP3}
⊗⊕IF F
THEN
  EVAL A
ELSE
  "CANCELADO" MSGBOX
⊗⊕

```

7. Escribe END para terminar la estructura de ejecución condicional.

```

RAD R42 HEX R= 'X'                                PRG
{HOME DRP3}
ELSE
  "CANCELADO" MSGBOX
END
⊗⊕
⊗⊕

```

Observa que después del END debes tener dos delimitadores de programa, el primero corresponde al subprograma con variable local F y el segundo cierra el programa principal.

8. Pulsa [ENTER] para salir del editor y cargar el programa en la pila,

```

RAD R42 HEX R= 'X'                                PRG
{HOME DRP3}
1: ⊗ "AREA TRAPECIO" {
  "X" "Y" "H" } { } {
  } { } INFORM → F ⊗
  IF F THEN EVAL A
  ELSE "CANCELADO"
⊗⊕

```

pulsa \uparrow [F1] para guardar la nueva versión con el nombre ATRAP.

¹⁷Puedes encontrar el comando MSGBOX en [PRG][OUT]. En la imagen he introducido retornos de carro y espacios para hacer el programa más legible.

9. Para finalizar vamos a verificar el funcionamiento del programa, ejecuta ATRAP y entra los valores de prueba.

```

AREA TRAPECIO
X 12.1
Y 4.7
H 5.8
EDIT CANCL OK
  
```

Acepta los valores pulsando [ENTER] o [OK], obtendrás:

```

RAD R&2 HEX R= 'X'
(HOME DRP8)
0:
4:
0:
2:
1: AREA: 48.7200
ATRAP A OK
  
```

10. Vuelve a ejecutar ATRAP,

```

AREA TRAPECIO
X
Y
H
EDIT CANCL OK
  
```

pero esta vez cancela el formulario, obtendrás:

```

AREA TRAPECIO
X
Y
H
CANCELADO
OK
  
```

Pulsa [ENTER], [F6] o [CANCEL], entonces se cierra el cuadro de diálogo y el programa finaliza sin error.

Actividad 6.7 Modifica el formulario de entrada del programa ATRAP para que asigne valores iniciales $X = 0, Y = 0, H = 0$.

Actividad 6.8 Crea un subdirectorio de HOME con el nombre DRP9. En ese directorio crea las funciones A y L que calculan el área y perímetro de un círculo a partir del radio

$$L = 2\pi R, \quad A = \pi R^2$$

y devuelven los resultado debidamente etiquetados. Una vez verificado el buen funcionamiento de las funciones A y L, construye el programa APC (Area y Perímetro del Círculo) que permite entrar el radio desde un formulario de entrada de datos y ejecutar las dos funciones A y L.

6.4 Uso avanzado del comando INFORM

En esta sección veremos como definir los *valores de reset*, las *especificaciones de formato* y los *formatos de campo*.

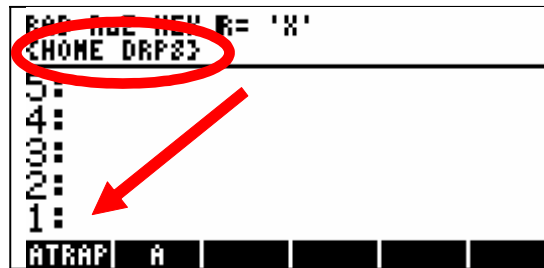
6.4.1 Valores de reset

Los *valores de reset* están definidos por la lista que ocupa el Nivel 2 en la entrada de INFORM. Si la lista no está vacía debe contener tantos valores como campos.

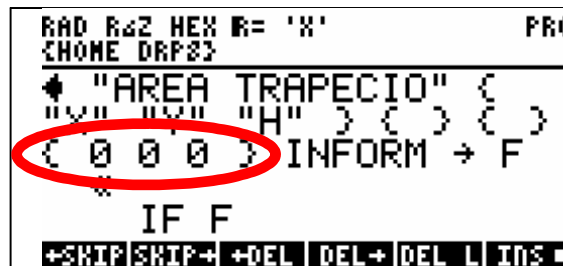
Vamos a modificar el programa ATRAP añadiendo los valores de reset.

Actividad 6.9 Realiza los siguientes pasos:

1. Accede al directorio DRP8



y pulsa \uparrow [F1] para recuperar el programa, pulsa \downarrow para activar el editor.



La lista marcada es la del Nivel 1, y contiene los valores iniciales. Si tienes una lista vacía, entra tres ceros.

2. La lista del Nivel 2

```

RAD R42 HEX R= 'X'          PRG
CHOME DRP83
* "AREA TRAPECIO" (
"X" "Y" "H" ) ( ) ( )
( 0 0 0 ) INFORM → F
*
IF F
+SKIP|SKIP+ +DEL |DEL+ |DEL L| INS ■

```

contiene los valores de reset, entra tres ceros también en esa lista.

```

RAD R42 HEX R= 'X'          PRG
CHOME DRP83
* "AREA TRAPECIO" (
"X" "Y" "H" ) ( )
( 0 0 0 ) * ←
( 0 0 0 ) INFORM → F
*
+SKIP|SKIP+ +DEL |DEL+ |DEL L| INS ■

```

3. Pulsa [ENTER] para cargar el programa en la pila

```

RAD R42 HEX R= 'X'          PRG
CHOME DRP83
1: * "AREA TRAPECIO" (
"X" "Y" "H" ) ( ) ( )
0 0 0 ) ( 0 0 0 )
INFORM → F * IF F
THEN EVAL A ELSE
ATRAP A | | | | |

```

y pulsa \uparrow [F1] para guardar la nueva versión con el nombre ATRAP.

4. Pulsa [F1] para ejecutar ATRAP, obtendrás

```

AREA TRAPECIO
X 0
Y 0
H 0
EDIT | | | | | CANCEL OK

```

Completa el formulario como sigue

AREA TRAPECIO	
X	41.12
Y	21.4
H	5.23
EDIT CANCL OK	

5. Pulsa la tecla [NEXT] para acceder a la segunda página de menú del formulario

AREA TRAPECIO	
X	41.12
Y	21.4
H	5.23
RESET CALC TYPES CANCL OK	



y pulsa [F1] para ejecutar la opción [RESET], obtendrás el siguiente cuadro de diálogo

AREA TRAPECIO	
X	41.12
Y	Reset value
H	Reset all
CANCL OK	

Si aceptas la opción Reset value, el campo seleccionado, en este caso el campo X, tomará el valor de reset definido en la lista de valores de reset. Si aceptas la opción Reset all, todos los campos tomarán el valor por defecto definido en la lista de valores de reset. Pulsa [OK] para aceptar Reset value, obtendrás

AREA TRAPECIO	
X	0
Y	21.4
H	5.23
RESET CALC TYPES CANCL OK	

Pulsa [NEXT], para volver a la primer página del menú del formulario

AREA TRAPECIO	
X	0
Y	21.4
H	5.28
EDIT CANCL OK	

6. Entra el valor 3.45 en el campo X,

AREA TRAPECIO	
X	3.45
Y	21.4
H	5.28
EDIT CANCL OK	

y acepta los valores, resulta

RAD R22 HEX R= 'X'	
[HOME DRP3]	
5:	
4:	
3:	
2:	
1:	AREA: 64.9828
ATRAP A	

6.4.2 Especificaciones de formato

Las *especificaciones de formato* permiten estructurar la presentación de los campos y las etiquetas en el formulario. Las especificaciones de formato son el objeto de Nivel 3 en la entrada de INFORM y pueden tener una de las formas siguientes:

- Una lista vacía { }.
- Una lista con el *número de columnas*: {NUM_COL}.
- Una lista con un par de números {NUM_COL TAB}, donde NUM_COL es el número de columnas y TAB es el número de espacios reservado para las etiquetas de campo, si no se especifica nada, INFORM toma los siguientes valores:

- NUM_COL = 1 (1 columna).
- TAB = 3 (tres espacios reservados para las etiquetas).

Actividad 6.10 Recupera el programa ATRAP y modifica la lista de especificaciones de formato de manera que el formulario tenga dos columnas de campos, esto es, debes sustituir la lista vacía del Nivel 3

```

RAD R42 HEX R= 'X'          PRG
CHOME DRP83
« "AREA TRAPEDIO" (
"X" "Y" "H" ) ( ) ( 0
0 0 ) ( 0 0 0 )
INFORM → F
«
<SKIP|SKIP+|+DEL|DEL+|DEL L|INS=

```

por {2}.

```

RAD R42 HEX R= 'X'          PRG
CHOME DRP83
« "AREA TRAPEDIO" (
"X" "Y" "H" ) (2) ( 0
0 0 ) ( 0 0 0 )
INFORM → F
«
<SKIP|SKIP+|+DEL|DEL+|DEL L|INS=

```

Guarda la nueva versión del programa y ejecútala, el formulario de entrada de datos tendrá ahora el siguiente aspecto

AREA TRAPEDIO	
X	0
Y	0
H	0

EDIT | | | CANCL | OK

Actividad 6.11 En esta actividad, vamos a ver el funcionamiento del parámetro TAB.

1. Recupera el programa ATRAP y establece el formato de una columna

```

RAD R42 HEX R= 'X'          PRG
(HOME DRP8)
« "AREA TRAPECIO" (
  "X" "Y" "H" ) ( 1 ) (
  0 0 0 ) ( 0 0 0 )
INFORM → F
»
+SKIP|SKIP+ +DEL|DEL+|DEL L|INS ▀

```

2. Cambia el nombre de tercer campo por ALTURA.

```

RAD R42 HEX R= 'X'          PRG
(HOME DRP8)
« "AREA TRAPECIO" (
  "X" "Y" "ALTURA" ) ( ...
  0 0 0 ) ( 0 0 0 )
INFORM → F
»
+SKIP|SKIP+ +DEL|DEL+|DEL L|INS ▀

```

Guarda la nueva versión del programa y ejecútalo, obtendrás el siguiente formulario:

```

AREA TRAPECIO
X 0
Y 0
ALTURA 0
EDIT | | | CANCEL OK

```

Observa que aparece una sola columna de campos. Por otra parte, la etiqueta del campo 3 es de longitud 6, eso provoca una mala alineación de los campos. Si estableces el valor de tabulación en 6, los campos estarán bien alineados.

3. Recupera el programa y modifica la lista de especificaciones de formato como sigue

```

RAD R42 HEX R= 'X'          PRG
(HOME DRP8)
« "AREA TRAPECIO" (
  "X" "Y" "ALTURA" )
  ( 1 6 ) ( 0 0 0 ) ( 0 ...
  ) INFORM → F
»
+SKIP|SKIP+ +DEL|DEL+|DEL L|INS ▀

```

Guarda la nueva versión del programa y ejecútalo, obtendrás el siguiente formulario:

```

AREA TRAPECIO
X      0
Y      0
ALTURA 0
EDIT  CANCL  OK
  
```

6.4.3 Formatos de campo

Recordemos que el comando INFORM toma como entrada 5 objetos:

- Nivel 5: “*Título del formulario*”, debe ser un string.
- Nivel 4: {*campo1, campo2, ...*}, lista de especificaciones de campos.
- Nivel 3: {*formato*}, lista de especificaciones de formato, puede ser una lista vacía.
- Nivel 2: {*lista de valores de reset*}, puede ser vacía.
- Nivel 1: {*lista de valores iniciales*}, puede ser vacía.

En esta sección nos ocupamos del objeto del Nivel 4, esto es, de la lista de *especificaciones de campo*.

$$\{ \text{campo1}, \text{campo2}, \text{campo3}, \dots \}.$$

Hemos visto anteriormente, que en su forma más simple, la especificación de campo está formada por un string que es la etiqueta del campo. En general un formato de campo puede tener una de las siguientes formas:

- “*etiqueta*”.
- { “*etiqueta*” “*texto de ayuda*” }.
- { “*etiqueta*” “*texto de ayuda*” *tipo1 tipo2 ... tipok* }.
- { }.

Si incluimos un “*texto de ayuda*”, éste texto se muestra en la base de la pantalla cuando seleccionamos el campo. Si especificamos uno o más *códigos de tipo*, el sistema verifica si el objeto entrado se ajusta a los tipos prescritos

y, en caso contrario, genera un mensaje de error. Algunos de los *códigos de tipo* más usuales se recogen en la tabla siguiente:

<i>Objeto</i>	<i>código de tipo</i>
Número real	0
Número complejo	1
Cadena de caracteres (string)	2
Matriz real	3
Matriz compleja	4
Lista	5
Nombre global	6
Programa	8
Objeto algebraico	9

Si usamos como especificación de formato de campo una lista vacía, entonces el campo anterior en la lista se expande ocupando el espacio correspondiente al campo con especificación { }.

Como primera actividad, vamos a ver el uso de texto de ayuda. Tomaremos como ejemplo de trabajo un formulario de entrada para el cálculo del área de un triángulo

$$A = \frac{B \cdot H}{2} \quad \text{donde} \quad \begin{cases} - A \text{ área.} \\ - B \text{ base.} \\ - H \text{ altura.} \end{cases}$$

Actividad 6.12 *Crea un subdirectorio de HOME con el nombre DRP9. Entra en DRP9 y realiza los siguientes pasos:*

1. *Entra los delimitadores de programa y escribe el título del formulario*

```

PRG
{HOME DRP9}
2:
1:
* "AREA TRIANGULO"
*
*

```

2. *A continuación vamos a entrar las especificaciones de los campos de entradas de datos, el programa tomará como entrada el valor de la base y la altura.*

```

RAD R42 HEX R= 'X'          PRG
[HOME DRP9]
1:
* "AREA TRIANGULO"
{ ("B" "BASE")
{ ("H" "ALTURA") } }
*

```

Observa que las especificaciones de los campos es una lista que contiene dos listas de la forma {“etiqueta” “texto de ayuda”}.

3. Completa la entrada de INFORM con una lista vacía para las especificaciones de formato del formulario (número de columnas y longitud de tabulación) y dos listas que asignan valores nulos como valores iniciales y valores de reset.

```

RAD R42 HEX R= 'X'          PRG
[HOME DRP9]
* "AREA TRIANGULO"
{ ("B" "BASE")
{ ("H" "ALTURA") } }
{} {0 0} {0 0}
*

```

Entra el comando INFORM y pulsa ENTER para salir del editor; guarda el programa con el nombre ATRG.

```

RAD R42 HEX R= 'X'          PRG
[HOME DRP9]
1: * "AREA TRIANGULO"
{ ("B" "BASE") } {
{ ("H" "ALTURA") } } {
} {0 0} {0 0}
INFORM *

```

4. Ejecuta el programa, obtendrás el siguiente formulario:

```

AREA TRIANGULO
B  [ ]
H  [ ]
BASE
EDIT  CANCL  OK

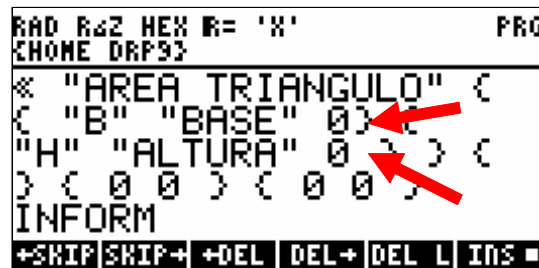
```

Observa como aparece el texto de ayuda al seleccionar los campos.



En la siguiente actividad vamos a completar la especificación de campos para que sólo acepten como valores números reales. En este caso debemos incluir el código 0 en las especificaciones de campo.

Actividad 6.13 Recupera el programa ATRG y modifica las especificaciones de campo como sigue



pulsa ENTER para salir del editor y cargar el programa en la pila.



Pulsa \uparrow [F1] para guardar la nueva versión del formulario. Ejecuta el programa y obtendrás:



Entra el valor 24 para la base

```

AREA TRIANGULO
B 24.
H 0
ALTURA
EDIT | | | CANCL | OK
  
```

y, a continuación, intenta entrar un nombre de variable para la altura,

```

AREA TRIANGULO
B 24.
H 0
X*
EDIT | | | CANCL | OK
  
```

pulsa ENTER y obtendrás el siguiente mensaje de error:

```

AREA TRIANGULO
B 24.
H 0
Bad Argument Type
X*
TYPES | | | OK
  
```

Pulsa [CANCEL] y entra el valor 12 para la altura,

```

AREA TRIANGULO
B 24.
H 12.
ALTURA
EDIT | | | CANCL | OK
  
```

pulsa [OK] para aceptar los valores, obtendrás

```

RAD R42 HEX R= 'X'
{HOME DPR9}
5:
4:
3:
2:      (24.0000 12.0000)
1:      1.0000
ATRQ

```

Para terminar, en la siguiente actividad, vamos a completar el programa ATRG, de forma que proporcione como salida el valor del área debidamente etiquetado.

Actividad 6.14 Realiza los siguientes pasos:

1. Escribe la siguiente función,

```

RAD R42 HEX R= 'X'
{HOME DPR9}
4:
3:
2:
1:  * → B H 'B*H/2'
    →NUM "AREA" →TAG *
ATRQ

```

y guárdala con el nombre A en el directorio {HOME DPR9}.

2. Recupera el programa ATRG y complétalo con una estructura de variable local que ejecute A cuando el usuario complete el formulario y escriba el mensaje "CANCELADO" cuando el usuario cancele el formulario. La estructura de variable local se muestra en las dos siguientes pantallas. Observa que en la segunda pantalla se repite una línea de la primera.

```

RAD R42 HEX R= 'X'
{HOME DPR9}
INFORM → F
*
  IF F
  THEN EVAL A
  ELSE "CANCELADO"
+SKIP|SKIP+ +DEL|DEL+|DEL L|INS

```

```

RAD R&Z HEX R= 'X'          PRG
[HOME DRP9]
      ELSE "CANCELADO"
MSGBOX
      END
  ✖
✖
+SKIP|SKIP+|+DEL|DEL+|DEL L|INS

```

3. Guarda la nueva versión con el nombre ATRG. Verifica el buen funcionamiento del programa.

7 Gestión de la memoria

7.1 Organización de la Memoria de la calculadora

La memoria de la calculadora está organizada en varias áreas. Si entras en la aplicación [FILES]

```

FILE MANAGER
0: IRAM      239KB
1: ERAM     252KB
2: FLASH    1021KB
HOME        240KB
HOME5
HOME1
HOME3
HOME2
CHDIR | | | | CANCL | OK

```

puedes observar que hay 4 áreas de memoria, que se denominan *HOME*, *Puerto 0*, *Puerto 1* y *Puerto 2*. Si tienes una HP49G+ o una HP50g, y has insertado una tarjeta SD, aparecerá además un *Puerto 3*, que corresponde a la tarjeta SD.

- En la zona *HOME* se guardan las variables de usuario y puede organizarse en directorios.
- El *Puerto 0* comparte la misma zona de memoria que HOME, en total unos 240 Kb. Si guardas objetos en el Puerto 0, pierdes espacio para HOME.
- El *Puerto 1* tiene un tamaño de unos 250 Kb, es independiente de HOME.
- El *Puerto 2* es una memoria de tipo flash y se mantiene aunque falle la alimentación. Tiene un tamaño aproximado de 1Mb en la Hp49G y de unos 800 Kb en la Hp49G+.

Debes tener en cuenta que los Puertos 0 y 1 son de acceso rápido, pero su contenido puede perderse si falla la alimentación; por el contrario, el contenido del Puerto 2 se mantiene aunque retires las baterías.

Actividad 7.1 *Accede al File Manager y observa la memoria disponible en HOME/Puerto 0. En el emulador que estoy usando se obtiene*

```

FILE MANAGER
0:IRAM      239KB
1:ERAM      252KB
2:FLASH     1021KB
HOME        239KB
PORT5
MED1
CIL3
CIL2
CHDIR      |      |      |      |
CANCL     OK

```

En este caso, dispongo de 239Kb libres para guardar objetos en el área HOME/Puerto 0.

Actividad 7.2 *Accede al editor de matrices y carga en la pila la siguiente matriz.*

```

DEG R&Z HEX R= 'X'
{HOME}
Z:
1:
      [ 1 2 3 4 ]
      [ 0 0 1 2 ]
      [ 3 2 3 6 ]
      [ 4 7 8 9 ]
EDIT VIEW STACK RCL PURGE CLEAR

```

Guárdala en la variable A. Accede al File Manager

```

FILE MANAGER
0:IRAM      239KB
1:ERAM      252KB
2:FLASH     1021KB
HOME        239KB
PORT5
MED1
CIL3
CIL2
CHDIR      |      |      |      |
CANCL     OK

```



y pulsa [F6] para entrar en la zona de memoria de HOME. Observa la variable A.

Memory: 243711 Select: 0	
{ }PPAR	LIST 74
LIST	NAME 85
DRPS	DIR 120
MED1	DIR 120
CIL3	DIR 170
CIL2	DIR 75
CILIN	DIR 285
MED	PROG 17
EDIT COPY MOVE RCL EVAL TREE	

Si pulsas [F1] puedes editar directamente la matriz en el editor de línea.

DEG R&Z HEX R= 'X'	PRG
{HOME}	
← [1 2 3 4]	
[0 0 1 2]	
[3 2 3 6]	
[4 7 8 9]]	
←SKIP SKIP+ +DEL DEL+ DEL L INS =	

Cambia, por ejemplo, el primer 1 por un 20

DEG R&Z HEX R= 'X'	PRG
{HOME}	
[[20 2 3 4]	
[0 0 1 2]	
[3 2 3 6]	
[4 7 8 9]]	
←SKIP SKIP+ +DEL DEL+ DEL L INS =	

y pulsa [ENTER] para salir del editor. Pulsa nuevamente [ENTER] para salir del File Manager. Si recuperas el contenido de A,

DEG R&Z HEX R= 'X'	PRG
{HOME}	
C:	
1:	[20 2 3 4]
	[0 0 1 2]
	[3 2 3 6]
	[4 7 8 9]]
A DRPS MED1 CIL3 CIL2 CILIN	

verás que se ha modificado el contenido de la matriz.

Para guardar un objeto en un puerto, puedes usar la opción COPY o MOVE de la aplicación *File Manager*. También puedes guardar el objeto directamente con el comando STO anteponiendo :num puerto: al nombre del objeto.

Actividad 7.3 Accede al File Manager y selecciona la matriz A que has creado en la actividad anterior.

Memory: 243874 Select: 0	
DIR	MATRIX 69
DIRP5	DIR 120
MED1	DIR 120
CIL3	DIR 170
CIL2	DIR 75
CIL1	DIR 285
MED	PROG 17
CASDIR	DIR 255

EDIT | COPY | MOVE | RCL | EVAL | TREE

Pulsa [F2] para ejecutar [COPY], aparecerá la siguiente pantalla,

PICK DESTINATION	
0: IRAM	237KB
1: ERAM	257KB
2: FLASH	1021KB
None	237KB
DIRP5	
MED1	
CIL3	
CIL2	

CANCL | OK

que te pide que selecciones un destino para la copia. Selecciona el Puerto 1

PICK DESTINATION	
0: IRAM	237KB
1: ERAM	257KB
2: FLASH	1021KB
None	237KB
DIRP5	
MED1	
CIL3	
CIL2	

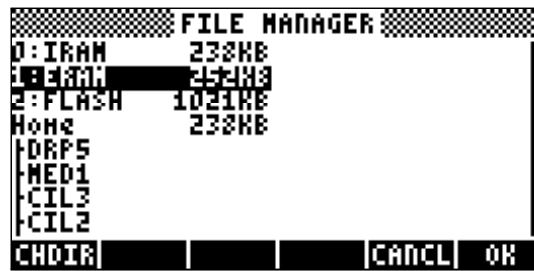
CANCL | OK

y pulsa [OK] o [ENTER], volverás a la vista HOME.

Memory: 243397 Select: 0	
DIR	MATRIX 69
DIRP5	DIR 120
MED1	DIR 120
CIL3	DIR 170
CIL2	DIR 75
CIL1	DIR 285
MED	PROG 17
CASDIR	DIR 255

EDIT | COPY | MOVE | RCL | EVAL | TREE

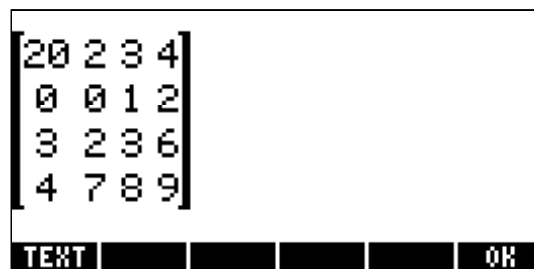
Pula [F6] para acceder al árbol de zonas de memoria y selecciona el Puerto 1.



Pulsa [ENTER] o [OK] y accederás a la vista del contenido del Puerto 1,



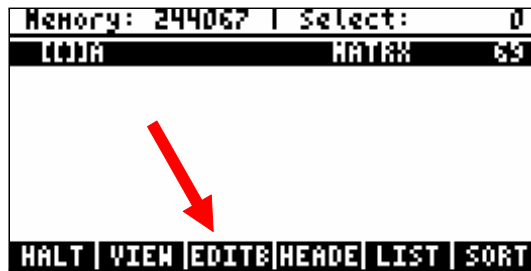
donde encontrarás la variable A que acabas de copiar. Pulsa [F1] para ejecutar la opción [VIEW] e inspeccionar el contenido de la variable.



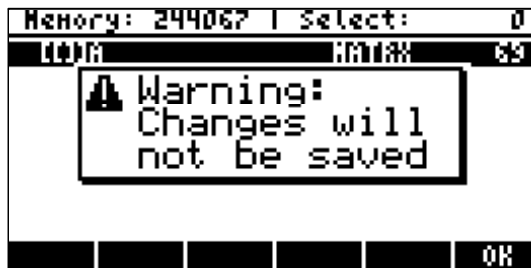
Pulsa [F6] para ejecutar [OK] y volver a la vista del contenido del Puerto 1. Si pulsas la tecla [NEXT] para ver las diferentes páginas de menú, obtendrás



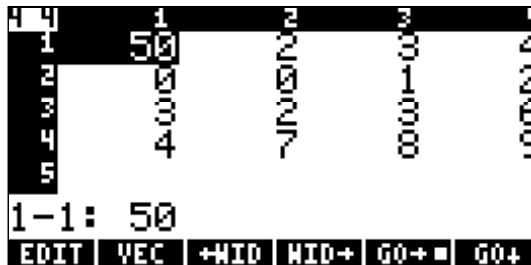
Pulsando nuevamente [NEXT], obtienes



Observa que la opción *EDIT* no aparece en los menús de herramientas de los Puertos. En cambio, sí que aparece la opción [EDITB], si la pulsas obtendrás el siguiente mensaje:



que te avisa de que los cambios que realices no se guardarán. Pulsa [OK] y cambia valor del elemento a_{11} de 20 a 50.



Pulsa [ENTER] para salir del editor.



Ejecuta [VIEW] y observa que, en efecto, los cambios no se han guardado.

También puedes ejecutar una programa almacenada en un puerto. En la siguiente actividad, vamos a crear un programa, almacenarlo en el Puerto 2 y, después, lo ejecutaremos desde la pila.

Actividad 7.5 Realiza los siguientes pasos.

1. Escribe el siguiente programa:

```
DEG R&2 HEX R= 'X'
{HOME}
-----
4:
3:
2:
1: « → X 'π*X^2' →NUM
   "AREA" →TAG »
A DRP5 MED1 CIL3 CIL2 CIL1
```

El programa toma el valor del radio de la pila, lo guarda en la variable local X y devuelve el valor del área del círculo de radio X debidamente etiquetado.

2. Escribe el nombre :2:CIR.

```
DEG R&2 HEX R= 'X'                                PRG
{HOME}
-----
3:
2:
1: « → X 'π*X^2' →NUM
   "AREA" →TAG »
:2:CIR
A DRP5 MED1 CIL3 CIL2 CIL1
```

y pulsa STO para guardar el programa en el Puerto 2 con el nombre CIR.

3. Accede al File Manager y verifica que el programa se ha guardado correctamente.

```
FILE MANAGER
0:IRAM 239KB
1:ERAM 255KB
2:MEMEM 1066KB
HOME 239KB
DRP5
MED1
CIL3
CIL2
CHOIR | | | CANCL | OK
```



Memory: 244098 Select: 0	
▢	CALC1 DIR 1653
▢	CALC2 DIR 1886
▢	CIR PRGM 50
VIEW COPY MOVE RCL EVAL TREE	

4. Hemos comentado anteriormente que no puedes editar un objeto guardado en los puertos, pero sí puedes acceder a su contenido y, si es un programa, ejecutarlo. Accede a la pila y carga el valor 3 y el nombre¹⁸
:2:CIR

DEG R42 HEX R= 'X'	
[HOME]	
0:	
1:	2:CIR
A DRP5 MED1 CIL3 CIL2 CIL10	

Tal como sucede con las variables, los programas guardados en los puertos no se evalúan automáticamente, pulsa [EVAL] para evaluar el programa, obtendrás:

DEG R42 HEX R= 'X'	
[HOME]	
0:	
1:	AREA:28.2743
A DRP5 MED1 CIL3 CIL2 CIL10	

7.3 Copias de seguridad del sistema

Los comandos ARCHIVE y RESTORE te permiten realizar copias de seguridad del contenido de tu calculadora. Supongamos que has realizado un buen número de las actividades propuestas en este tutorial, entonces habrás creado varios directorios y al pulsar [VAR] aparecerá un menú con varias etiquetas de directorio.

¹⁸Observa que cuando cargas :2:ACIR en la pila, los dos puntos iniciales no se muestran.



Este es un buen momento para hacer limpieza, pero quizás sería recomendable guardar los programas realizados en lugar de borrarlos. Los comandos ARCHIVE y RESTORE permiten realizar una copia de seguridad del contenido del área HOME de la calculadora. Puedes encontrar los comandos ARCHIVE y RESTORE en la segunda página del menú [PRG][MEM]



o en el catálogo de comandos.



Si tienes información valiosa en tu calculadora, quizás prefieras realizar las actividades de esta sección con el emulador.

En la siguiente actividad, vamos a guardar una copia de seguridad denominada SEG1 en el Puerto 2.

Actividad 7.6 Realiza los siguientes pasos:

1. Escribe el nombre del objeto de seguridad.

```

RAD R&Z HEX R= 'X'          PRG
[HOME]
4:
00:
00:
1:
2: SEG1
DRP9 | DRP8 | DRP7 | DRP6 | DRP5 | MED1

```

Pulsa [ENTER] para cargarlo en la pila y ejecuta el comando ARCHIVE.

```

RAD R&Z HEX R= 'X'          PRG
[HOME]
4:
00:
00:
1:
2: SEG1
ARCHI|RESTO| | | | PRG

```

2. Después de ejecutar ARCHIVE, accede a la aplicación File Manager y observa el contenido del Puerto 2.

```

FILE MANAGER
0: IRAM      237KB
1: ERAM      255KB
2: SEG1     2015KB
HOME        237KB
DRP9
DRP8
DRP7
DRP6
CHDIR | | | CANCL | OK

```

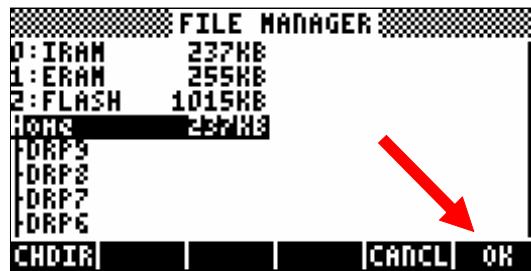
Verás que se ha creado el objeto de seguridad SEG1.

```

Memory: 241884 | Select: 0
CALC1  DIR  1653
CALC2  DIR  1886
CIR    PROG  50
SEG1  DIR  2015
VIEW | COPY | MOVE | RCL | EVAL | TREE

```

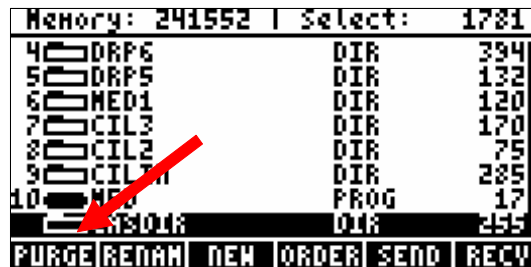
3. Accede al File Manager y entra en HOME.



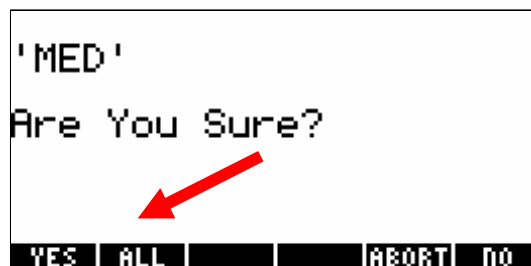
Pulsa reiteradamente [ENTER] para marcar todos los objetos y directorios de HOME excepto el directorio CASDIR.



Pulsa [NEXT] para acceder a la segunda página del menú del File Manager



y pulsa [F1] para ejecutar [PURGE], obtendrás la siguiente pantalla:

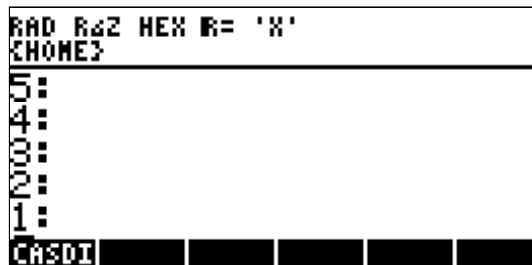


Pulsa [F2] para ejecutar la opción [ALL], obtendrás:



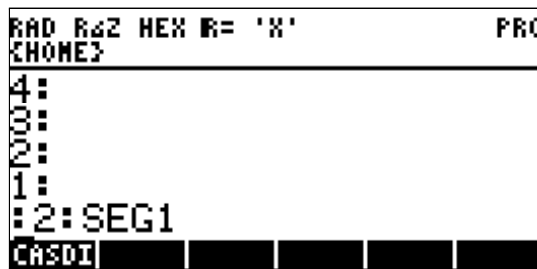
Es decir, acabas de borrar todo el contenido de HOME excepto el directorio CASDIR.

4. Pulsa [CANCEL] para salir del File Manager y pulsa [VAR], obtendrás:



que te confirma que has borrado todo el contenido de HOME.

5. Escribe el nombre del objeto de seguridad



y ejecuta el comando RESTORE.



Verás como se reinicia el sistema, pulsa [VAR] para acceder al área de variables y obtendrás:

```
RAD R&Z HEX R= 'X'  
{HOME}  
5:  
4:  
00:  
0:  
1:  
DRP9 | DRP8 | DRP7 | DRP6 | DRP5 | MED1
```

es decir, se ha recuperado el contenido de HOME guardado en SEG1.

Importante Debes tener en cuenta que la ejecución de RESTORE borra todo el contenido actual de HOME.