```c
/*-////////////////////////////////////////////////////////////////////////
// chksep.c, ver 1.0
// Contents: Muestra el nº de las lineas con +/- separadores que la primera.
// Entorno:  \djgpp\entorno.bat
// Compilar: gcc chksep.c -o chksep.exe
////////////////////////////////////////////////////////////////////////-*/
#include <stdio.h>
#include <string.h>

#define FALSE   0
#define TRUE    1
#define SEPDEF 59
#define EOL    10


/*-////////////////////////////////////////////////////////////////////////-*/
   int main(argc, argv)

/* PURPOSE: Muestra el nº de las lineas con +/- separadores que la primera.
////////////////////////////////////////////////////////////////////////////-*/
  int argc;
  char *argv[];
{
  int  sepChr;    /* Ascii del separador como entero */
  char sepStr[2]; /* String de 2 posiciones para visualizar el separador */

  int  sepCnt;    /* Contador de separadores de cada linea */
  int  sepFst;    /* Contador de separadores de la primera linea */

  int  linCnt;    /* Contador de lineas del fichero */

  int  chrInp;    /* Para leer caracteres del standard input */

  sepStr[1] = 0;  /* Poner el fin del separador */

  sepCnt    = 0;  /* Inicializar a cero el contador de separadores */
  sepFst    = 0;  /* Separadores de la 1ª linea, referencia para el resto */

  linCnt    = 1;  /* Comenzamos por la primera linea */


/*-////////////////////////////////////////////////////////////////////////
// Ayuda:
////////////////////////////////////////////////////////////////////////-*/
  if((argc!=2))
  {
    printf(
      "\nMuestra el numero de las lineas con +/- separadores que la primera"
      "\nejemplo: chksep.exe 42 < mifichero.txt"
      "\nargumento: 42 es el ascii del separador (42=[*], 59=[;],...),"
      "\n           trabaja con separadores de un solo caracter y "
      "\n           del codigo ascii 1 al codigo ascii 255."
      "\n           Si el argumento es erroneo asume [;] como separador."
      "\n           Lee del standard input y escribe en el standard output,"
      "\n           con [< mifichero.txt] lee del fichero [mifichero.txt]."
      "\nVertice Sistemas S.L. (2008)\n");
  }
/*-////////////////////////////////////////////////////////////////////////
// Proceso:
////////////////////////////////////////////////////////////////////////-*/
```

```
61      else
62      {
63        sscanf(argv[1],"%d", &sepChr); /* Leer el codigo ascii de entrada */
64        if(sepChr < 1 || sepChr > 255) /* Si no es un ascii normal */
65        {
66          sepChr = SEPDEF;
67          printf("\nArgumento [%s] incorrecto, se asume [%d].", argv[1], sepChr);
68        }
69
70        sepStr[0] = sepChr; /* Solo para visualizar */
71        printf("\nSeparador = [%s].", sepStr);
72
73        while((chrInp=getc(stdin)) != EOF)
74        {
75               if(chrInp==sepChr) { sepCnt++; } /* Es un separador */
76          else if(chrInp==EOL)                  /* Es un salto de linea */
77          {
78            if(linCnt==1)                       /* Estabamos en la 1ª linea */
79            {
80              sepFst = sepCnt; /* Referencia para el resto de lineas */
81              printf("\n[%d] [%s] en la primera linea.", sepFst, sepStr);
82            }
83            else                                /* Linea normal */
84            {
85              if(sepFst!=sepCnt)                /* Faltan o sobran separadores */
86              {
87                printf("\nlinea %d: [%d] [%s] en vez de [%d].",
88                       linCnt, sepCnt, sepStr, sepFst);
89              }
90            }
91            linCnt++;         /* Incrementar el contador de lineas */
92            sepCnt = 0;       /* Poner a cero el contador de separadores */
93          }
94        }
95        printf("\nProcesadas [%d] lineas.", linCnt);
96      }
97    }
98
99
100
101
```