```
/*
 * images
 */

/*
 * Synopsis: images < unix_PS_file > unix_PS_file.images_split
 */

/*
 * Each image, which is a string of hexadecimal digits surrounded by < and
 * > is made to begin on a new line and is split into chunks, all but the
 * last of which is of length 72. The text after the end of the image
 * comes after a new line. The syntax of images specifies that besides
 * white space, only the digits 0-9 and the letters A-F and a-f are legal
 * inside the image and that white space is not significant. The program
 * does not check that only legal characters are inside the image, but it
 * does take advantage of nonsignificance of white space to split images
 * into chunks that fit on a single editable line.
 */

#include <stdio.h>
#define LIMIT 72

main()
{
        char curr;
        int i; /* position of current image character in its output line */

        /* skip through to find an image.. */
        curr = getchar();

        while (curr != EOF) {

                switch (curr) {

                        case('<'): {
                                /* move the beginning of an image to the
                                next output line and begin counting image
                                characters */
                                printf("\n%c", curr);
                                i=0;
                                if ((curr = getchar()) != EOF) {
                                        exit(0);
                                }
                                /* every LIMIT characters in image, insert
                                a new line */
                                while (curr != '>') {
                                        if (i < LIMIT) {
                                                printf("%c", curr);
                                                i += 1;
                                        } else {
                                                printf("\n%c",curr);
                                                i=0;
                                        }
                                        if ((curr = getchar()) != EOF) {
                                                exit(0);
                                        }
                                }
                                break;
                        }

                        default: {
                                /* skipping through */
                                printf("%c", curr);
                                break;
                        }
```

```
		} /* end switch */

		curr = getchar();
	} /* end while */

	exit(0);
}
```