lab09

```
$ gcc lab09.c

$ ./a.out 69 < US2981877.tex
                        Patent Number: 2,981,877
                 SEMICONDUCTOR DEVICE-AND-LEAD STRUCTURE
                 Robert N. Noyce, Los Altos, California
     Assignor to Fairchild Semiconductor, Mountain View, California
  5            Filed July 30, 1959, Serial Number 830,507
                        10 Claims. (Cl. 317-235)

    This invention relates to electrical circuit structures_
    This invention relates to electrical circuit structures
    incorporating semiconductor devices. Its principal objects are_
    incorporating semiconductor devices. Its principal objects are
 10 these: to provide improved device-and-lead structures for making_
 10 these: to provide improved device-and-lead structures for making
    electrical connections to the various semiconductor regions; to_
    electrical connections to the various semiconductor regions; to
    make unitary circuit structures more compact and more easily_
    make unitary circuit structures more compact and more easily
    fabricated in small sizes than has heretofore been feasible; and_
    fabricated in small sizes than has heretofore been feasible; and
    to facilitate the inclusion of numerous semiconductor devices_
    to facilitate the inclusion of numerous semiconductor devices
    In brief, the present invention utilizes dished junctions_
    In brief, the present invention utilizes dished junctions
    extending to the surface of a body of extrinsic semiconductor, an_
    extending to the surface of a body of extrinsic semiconductor, an
    insulating surface layer consisting essentially of oxide of the_
    insulating surface layer consisting essentially of oxide of the
 20 same semiconductor extending across the junctions, and leads in_
 20 same semiconductor extending across the junctions, and leads in
    ......
    ......
    References Cited in the file of this patent UNITED STATES PATENTS_
    References Cited in the file of this patent UNITED STATES PATENTS
605 2,813,326 Liebowitz Nov. 19, 1957 2,836,878 Shepard June 3, 1958_
605 2,813,326 Liebowitz Nov. 19, 1957 2,836,878 Shepard June 3, 1958
```

```
score: 82.0
o. [Output] Program output format is incorrect
o. [Coding] lab09.c spelling errors: paragragh(1)
o. [Format] Program format can be improved.
o. [Efficiency] can be improved.
```

lab09.c

```c
// EE231002 Lab09. Word Processing
// 110060007, 黃俊穎
// 2021/12/06

#include <stdio.h>
#include <string.h>
#include <stdlib.h>

char PARA[1500];    // an input paragraph
int LN = 0;         // line number of printed text
int LW;             // line width of output lines

void line(void);     // print out line number
int width(int pos, int para_pos);     // calculate line width

int main(int argc, char *argv[])
{
    int count;       // counter of words in six title lines
    int i, j;        // variables in loops
    int pos = 0;         // line position
    int para_pos = 0;    // position in paragragh

    LW = atoi(argv[1]) - 4; // line width except for line spaces

    for (i = 0; i < 6; i++) {
        line();      // print out line number
        count = 0;   // initialize counter
        // store each line's word, and calculate the words
        for (j = 0; (PARA[j] = getchar()) != '\n'; j++, count++);
        PARA[j] = '\0';        // let change line to be '\0'
        // set the words to center
        for (j = 0; j < (LW - count) / 2; j++) {
            printf(" ");       // print out space before the words
        }
        printf("%s\n", PARA);  // print out the title
    }

    while ((PARA[0] = getchar()) != EOF) {  // if getchar doesn't finish
        line();                // print out line number
        if (PARA[0] == '\n') {
            printf("\n");      // change line if read empty line
        } else {
            pos = 0;           // initialize new line and paragraph's
            para_pos = 0;      // position
            for (j = 1; (PARA[j] = getchar()) != '\n'; j++);
            // save each line's word
            // run the loop from 0 to change line
            for (i = 0; PARA[i] != '\n'; i++) {
                printf("%c", PARA[i]);   // print out paragraph
                pos++;             // calculate line position
                para_pos++;        // calculate paragraph position
                // if paragraph is space, judge line width
                if (PARA[i] == ' ') {
                    pos = width(pos, para_pos);
                    // execute the line width's function
                }
            }
            printf("\n");
        }
    }
    return 0;
}

// calculate line numbers and print out
void line(void)
{
    LN++;                   // calculate line numbers
    if (LN % 5 == 0)        // print line number if line number is
        printf("%3d ", LN);  // multiple of five
    else
        printf("    ");     // otherwise, print 4 spaces
}

// to judge the line width if it's over the rule
int width(int pos, int para_pos)
{
    int i, j = pos;         // i is for loop, j is to judge if it's over

    // run the loop to space or change lines
    for (i = para_pos + 1; PARA[i] != ' ' && PARA[i] != '\n'; i++) {
        j++;                // calculate j
    }
    if (j < LW)    // if j < line width, we can continue in same line
        return pos;         // return line position
    else
        printf("\n");       // change line
        line();             // print out line numbers
```

```
    line();             // print out line numbers
```

```
        return 0;           // let line position to 0
```

```
    return 0;           // let line position to 0
```

```
}
```