lab09

```
$ gcc lab09.c
$ ./a.out 76 < ../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 incorporating
    semiconductor   devices.   Its   principal objects are these: to provide
 10 improved device-and-lead structures for making electrical connections to
    the   various  semiconductor regions; to make unitary circuit structures
    more   compact   and   more   easily  fabricated in small sizes than has
    heretofore   been  feasible; and to facilitate the inclusion of numerous
    semiconductor devices within a single body of material.
...
...
...
    10.  A semiconductor device comprising a body of extrinsic semiconductor
    having   a   surface,   said  body containing adjacent P-type and N-type
    regions   with   a dished junction therebetween, said junction having an
545 edge   that extends to said surface and there forms an elongated, closed
    figure,   first and second contacts in the form of parallel metal strips
    adherent   to   said surface, said first contact being wholly within and
    said   second   contact   wholly   without said edge of the junction, an
    insulating   layer   consisting   of oxide of said semiconductor on said
550 surface   and extending across said junction, and a metal strip adherent
    to said insulating layer and extending thereover across said junction to
    connect physically and electrically with said first contact.

    References   Cited   in   the  file of this patent UNITED STATES PATENTS
555 2,813,326   Liebowitz   Nov.   19,   1957 2,836,878 Shepard June 3, 1958
    2,842,723 Koch et al. July 8, 1958 2,849,664 Beale Aug. 26, 1958
```

Output is correct

CPU time: 0.0094486 sec

score: 92

o. [Output] Program output is correct, good.

```
o. [Format] Program format can be improved
```

lab09.c

```
// EE2310 lab09 Word Processing
// 109061217, 林峻霆
// Date: 2020/11/30

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

char PARA[1500];                                // store what we read
int LN = 0;                                     // amount of lines
int LW;                                         // line width

int To_int(char s[2]);                          // function change char to int
void Print_title();                             // function print title
void Print_title(void);                             // function print title
void Print_Para();                              // function print paragraph
void Print_Para(void);                              // function print paragraph

int main(int argc, char *argv[])
{
    LW = To_int(argv[1]);                       // read LW from command
    Can use atoi function.
    Print_title();                              // print the title
    Print_Para();                               // print the whole paragraph

    return 0;                                   // end the program
}

void Print_title()
void Print_title(void)
    // The function that print the title
    //      step 1. Input the title from text
    //      step 2. Separate them by '\n', also record amount of line
    //      step 3. according to the information we got, print result
{
    int i = 0;                                   // PARA's index
    int j = 0;                                   // parameter for loop
    int k, t;                                    // parameter for loop
    int block;                                   // amount of blank space

    PARA[0] = getchar();                         // input a char
    do {                                         // continue input char until
        i++;                                     // met a double '\n'
        PARA[i] = getchar();
    } while (PARA[i] != '\n' || PARA[i - 1] != '\n');

    while(i > 0) {
    while (i > 0) {
        k = j;                                    // store initial j
        for (; PARA[j] != '\n'; j++);             // count until met '\n'
        LN += 1;                                  // increase LN
        if (LN % 5 == 0)                          // check need to print
            printf("%3d ", LN);                   // LN or four spaces
        else
            printf("    ");

        if (LW - 5 - j + k >= 0) {                // if not exceed LW
            block = (LW - 4 - j + k) / 2;         // calculate block amount
            for (t = 0; t < block; t++)           // print the block
                printf(" ");
            for (; k <= j; k++) {                 // print the title
                i -= 1;
                printf("%c", PARA[k]);
            }
            j = j + 1;
        }
        else {                                    // if exceed LW
            i -= 1;
            for (t = j; PARA[t] != ' '; t--);     // separate last few words
            block = LW - 4 - t + k;               // calculate block
            for (; k < t; k++) {
                i -= 1;
                if (PARA[k] == ' ' && block > 1) { // print the block and the
                    block -= 2;                    // words
                    printf("  %c", PARA[k]);
                }
                else if (PARA[k] == ' ' && block == 1) {
                    block -= 1;
                    printf(" %c", PARA[k]);
                }
                else
                    printf("%c", PARA[k]);
            }
            printf("\n");                          // print next line
            LN += 1;                               // increase LN
            if (LN % 5 == 0)                       // check whether print LN
                printf("%3d ", LN);
            else
                printf("    ");
            block = (LW - 4 - j + t) / 2;
            while (block--)                        // print the remain
                printf(" ");
            for (t = t + 1; t <= j; t++) {
                i -= 1;
                printf("%c", PARA[t]);
            }
            j = j + 1;                             // change j to j + 1
        }
    }
    LN++;                                          // increase LN
    printf("\n");                                  // print next line
}

int To_int(char s[2])
    // The function that change a string into integer
    //      step 1. Compare each digit, store it in integer in
    //              another array.
    //      step 2. Calculate the value and return.
{
    int i = 0;                                     // parameter for loop
    int j;                                         // parameter for loop
    int find;                                      // parameter for break
    int num[2];                                    // array to store number

    while (i < 2) {
        find = 1;
        for (j = 0; j < 10 && find; j++) {         // check for every digits
            if (s[i] - j == '0') {
                num[i] = j;
                find = 0;
            }
        }
        i += 1;
    }
    j = 10 * num[0] + num[1];                      // calculate the result
    return j;                                      // return the result
}

void Print_Para()
void Print_Para(void)
    // Print the paragraph by separating them with '\n' and end when read EOF
    //      step 1. keep reading until the next string will excess LW
    //      step 2. rearrange the already-read string and print
    //      step 3. continue read until met next line command
    //      step 4. check if the text end (met EOF)
    //                  if yes, print the remain PARA.
    //                  if no, print the remain and read next paragraph.
{
    int i;                                         // parameter for loop
    int j;                                         // parameter for loop
    char l;                                        // detect next paragraph
    char c[30];                                    // array to store string
    int block;                                     // amount of block at end
    int find = 1;                                  // parameter for break;
    int detect = 1;                                // parameter for detect;

    scanf("%s", PARA);                             // input string
    scanf("%s", c);                                // input string
    while (find) {                                 // check line is full or not
        if (strlen(PARA) + strlen(c) + 1 > LW - 4) {
            LN += 1;                               // increase LN
            if (LN % 5 == 0)                       // check print LN or
                printf("%3d ", LN);                // four spaces
            else
                printf("    ");
            block = LW - 4 - strlen(PARA);         // calculate block amount
            for (j = 0; j < strlen(PARA); j++) {   // print blocks and words
                if (PARA[j] == ' ') {
                    for (i = j + 1; i < strlen(PARA) && detect; i++) {
                        if (PARA[i] == ' ')
                            detect = 0;
                    }
                    if (detect) {
                        while (block--)
```

```
                            printf(" ");
                        }
                        else if (block > 1) {
                            detect = 1;
                            block -= 2;
                            printf("  ");
                        }
                        else if (block == 1) {
                            detect = 1;
                            block -= 1;
                            printf(" ");
                        }
                    }
                    printf("%c", PARA[j]);             // print the content
                }
                printf("\n");                          // print next line
                strcpy(PARA, c);                       // copy c to y
            }
            else {
                strcat(PARA, " ");                     // combine " " and c
                strcat(PARA, c);                       // with PARA
            }

            l = getchar();                             // read next char
            if (l == '\n') {                           // check if paragraph end
                LN += 1;                               // increase LN
                if (LN % 5 == 0)                       // check to print LN or
                    printf("%3d ", LN);                // four spaces
                else
                    printf("    ");

                printf("%s\n", PARA);

                l = getchar();                         // read next char
                if (l == EOF)                          // check if end the text
                    find = 0;
                else {
                    LN += 1;
                    if (LN % 5 == 0)
                        printf("%3d \n", LN);
                    else
                        printf("\n");
                    scanf("%s", PARA);                 // read next input
                    scanf("%s", c);                    // read next input
                }
            }
            else
                scanf("%s", c);                        // read next input
        }
    }
```