lab10

```
// EE231002 Lab10. Word Processing
// 108061213, 劉奕緯
// Nov. 27, 2019

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

#define PN 1000                 // max #character for a paragraph
#define PW 80                   // Page width
#define LN 52                   // max #line for output buffer
#define CW 38                   // column width

// function to process a paragraph (*str) and store in Buffer
// where Buffer has n string
void ProPara(char *str, char Buffer[LN][CW + 1], int *n);
// funtion to output paragraph by its Buffer
void OutputPara(char Buffer[LN][CW + 1], int n);

int main()
int main(void)
{
    char para[PN + 1];          // a sting of paragraph add a space for '\0'
    char outBuffer[LN][CW + 1]; // buffer for output
                                // since i will process it by string
                                // add a space for '\0'
    int i = 0, j;               // index
    int n;                      // #row of buffer
    int blank;                  // #blank space need to print

    j = 3;                      // process the first three lines
    while (j--) {
        while((para[i] = getchar()) != '\n') i++;
        while ((para[i] = getchar()) != '\n') i++;
        para[i] = '\0';
        blank = (PW -  strlen(para)) / 2;
        while (blank--) printf(" ");
        printf("%s\n", para);
        i = 0;                  // next line
    }
                                // process remain paragraph
    while ((para[i] = getchar()) != EOF) {
        if (para[i] == '\n') {  // we had got whole paragraph
            para[i] = '\0';     // to become a string
            if (strlen(para) == 0)
                printf("\n");   // if paragraph is void, just print "\n"
            else {              // else print the paragraph in required form
```

```
                ProPara(para, outBuffer, &n);
                OutputPara(outBuffer, n);
            }
            i = 0;              // next paragraph
        }
        else i++;               // next letter
    }
    return 0;
}

// function to out put a paragraph (*str)
void ProPara(char *str, char Buffer[LN][CW + 1], int *row)
{
    char word[CW];              // buffer for a word
    int i, j, sum;              // index, index,  #letter in a column
    int done = 0;               // done become 1,when string end

    *row = i = j = sum = 0;     // initialize
    while(!done) {
    while (!done) {
        if ((word[i] = *(str++)) == '\0')done = 1;
        if (word[i] == ' ' || word[i] == '\0') {
            word[i] = '\0';     // got a word
            if (sum + 1 + i > CW) {
                                // if it cannot put in column
                                // put it to next row
                strcpy(Buffer[++j], word);
                sum = i;        // now # of letter in this row is sum
            }
            // Note that fiist word of first row is in following case,
            // while the first word of else rows is in the former.
            else {              // if it can put in column
                                // add a blank space and put this word in
                if(j + sum > 0) {
                if (j + sum > 0) {
                    strcat(Buffer[j], " ");
                                // no blank for first word of first row
                    sum++;      // a blank sapce
                }
                strcat(Buffer[j], word);
                sum += i;       // add a word(a word has i letters)
            }
            i = 0;              // next word
        }
        else i++;               // got a letter
    }
    *row = j;
}

// funtion to output paragraph by its Buffer
```

```
void OutputPara(char Buffer[LN][CW + 1], int n)
{
    int i, j, b = 0;

    for(i = 0, j = n / 2 + 1; i < n / 2 + 1; i++, j++) {
    for (i = 0, j = n / 2 + 1; i < n / 2 + 1; i++, j++) {
        b = CW - strlen(Buffer[i]);
        while (b--) printf(" ");
                                // print pre- blank
        printf("%s | %s\n", Buffer[i], Buffer[j]);
                                // print a line
    }
    for (j = 0; j < LN; j++) strcpy(Buffer[j],"");
    for (j = 0; j < LN; j++) strcpy(Buffer[j], "");
                                // clear this buffer
}
```

[Format] can be improved.
[Coding] lab10.c spelling errors: fiist(1), funtion(2), pre(1), sapce(1)
[Efficiency] can be improved.

Score: 77