1 将《The\_Holy\_Bible.txt》中的标点符号替换成空格， 大写字母转换成小写字母并将处理 过的文本保存到 “The\_Holy\_Bible\_Res.txt”中。

int main()

{

FILE \*fp=fopen("D:\\The\_Holy\_Bible.txt","r+");

FILE\* fpw = fopen("D:\\The\_Holy\_Bible\_Res.txt", "w+");

if(NULL==fp)

{

perror("fopen");

return;

}

char buf[1024];

int ret, len;

ret = ftell(fp);

while (fgets(buf, sizeof(buf), fp) != NULL)

{

len = strlen(buf);

printf("改前：\n %s\n", buf);

for (int i = 0; i < len; i++) {

if (buf[i] >= 65 && buf[i] <= 90) {

printf(" big wright is %c\n", buf[i]);

buf[i] += 32;

}

else if (buf[i] <= 47 && buf[i] >= 33 && buf[i] >= 58 && buf[i] <= 64) {

printf(" 标点符号 is %c\n", buf[i]);

buf[i] = ' ';

}

}

printf("改后： \n");

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

int retw;

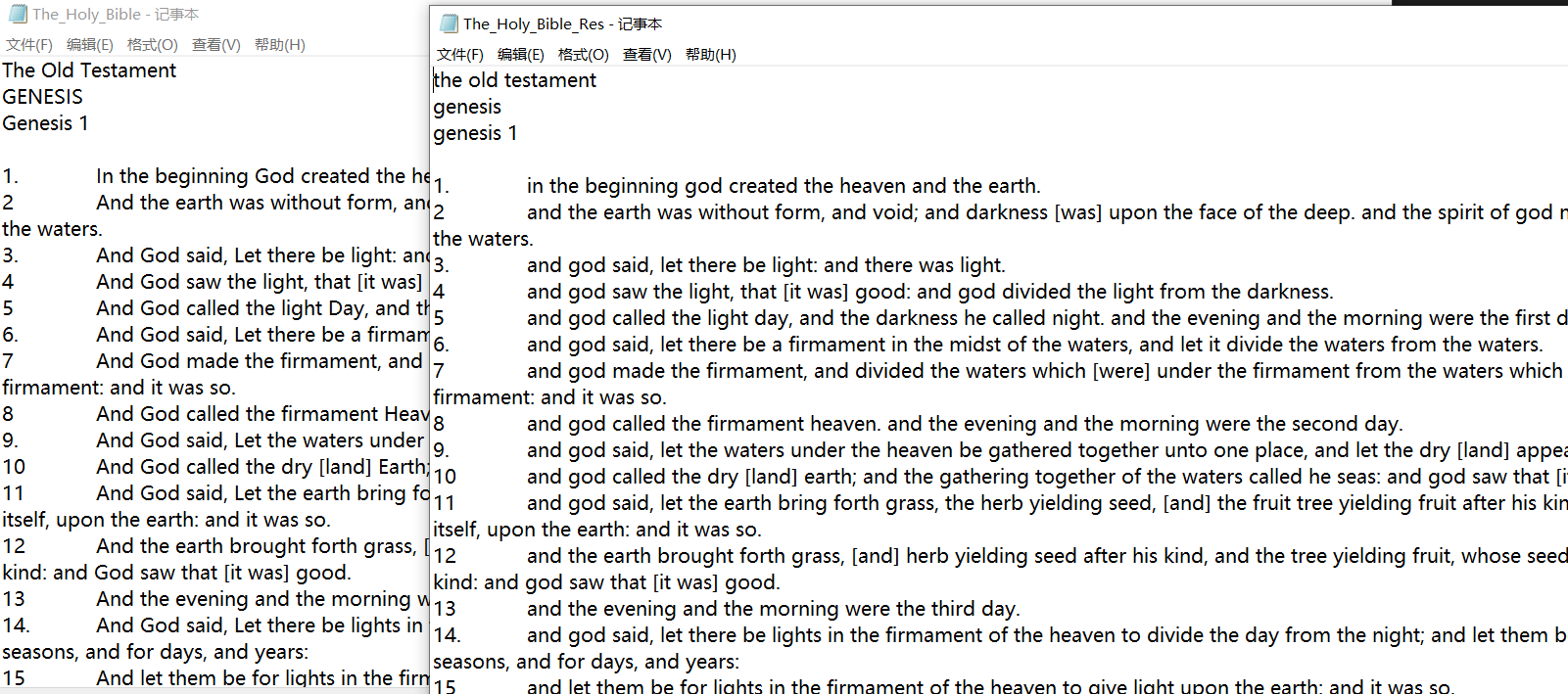
fputs(buf, fpw); //写入

fclose(fp);//关闭文件指针

fclose(fpw);//关闭文件指针

system("pause");

}



2统计” The\_Holy\_Bible\_Res.txt “ 中字符的个数，行数，单词的个数，统计单词的词频并打 印输出词频最高的前 10 个单词及其词频。

#define Word\_Max 20

#define StringN 500

struct word {

char w[Word\_Max];

int count;

}a[StringN+1];

int main(void) {

FILE\* fp = fopen("D:\\The\_Holy\_Bible.txt", "rb"); //用于读取的二进制文件

int i=0,j;

int k = 0; // k专门用来加一新加入的结构体位置

int flag = 0;

char word[20];

int charN = 0, hangN = 0, stringN = 0; //字符数，行数，单词数

// 以字符为单位，一趟匹配

// 如果还没读到非字符

// if NULL

while (!feof(fp))

{

char ch = tolower(fgetc(fp)); // 全部转化为小写

if (ch=='\n') hangN++;

if (isalpha(ch)) {

word[i] = ch;

i++;

charN++;

}

else // 读到了非字符，则字符集齐单词

{

word[i] = '\0'; // 将单词个体封装好

stringN++;

for (j = 0; j <= k; j++) { //在结构体中一趟匹配单词

if (strcmp(a[j].w, word) == 0) {

a[j].count++; //词频加一

flag = 1; //存在此单词的flag置1

break;

}

}

if(flag == 0) { // 若单词第一次出现(flag==0)，新单词加入结构体中

strcpy(a[k].w, word); // 此处不能a[k].w = word

a[k].count = 1; //出现一次。只有这一句写入结构体

k++;

}

flag = 0; // 维护flag

i = 0; // 维护i

}

}

//按照词频从小到大排序

for (int i = 0; i < k; i++) {

for (int j = i + 1; j < k; j++) {

if (a[i].count <= a[j].count) {

a[StringN] = a[i];

a[i] = a[j];

a[j] = a[StringN];

}

}

}

printf(" 单词数：%d次\n",stringN);

printf(" 字符数：%d次\n", charN);

printf(" 行数：%d行\n\n\n", hangN);

printf(" 词频最高的10个单词依次是\n");

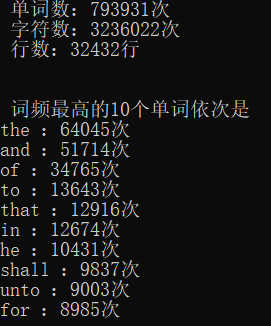
for (int i = 0; i < 10; i++) {

printf("%s ：%d次\n", a[i].w, a[i].count);

}

return 0;

}



3 使用计数排序给 1 亿个数排序。测试排序时间

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#define N 100000000

#define M 100

void arrCount(int\* arr) {

int count[M] = { 0 };

int i, j, k;

// 遍历数组后，统计数组中每个元素出现次数

for (i = 0; i < N; i++) {

count[arr[i]]++;

}

//k用来记录数组中哪些元素已经填入值

k = 0;

//将每个数值出现的次数，依次从前到后填入数组arr

for (i = 0; i < M; i++) {

for (j = 0; j < count[i]; j++) {

arr[k++] = i;

}

}

}

int main()

{

int\* arr = (int\*)malloc(sizeof(int) \* N); // 为数组生成所需全部空间

int i;

time\_t start, end; // ----------生成随机数

srand(time(NULL));

for (i = 0; i < N; i++)

{

arr[i] = rand() % M;

}

printf("计数排序结果为： \n");

start = time(NULL); //------------十个随机数生成完毕

arrCount(arr);

end = time(NULL);

printf("use time=%d\n", end - start);

system("pause");

}



4有一个记录学生信息的文件，每一行记录一名学生的信息，格式入下 学号\t 姓名\t 性别\t 分数 1\t 分数 2\t 分数 3\n. 要求：（1）读取文件的内容，串成一个链表。 （2）按照总分递减排序将结果保存到原文件。

typedef struct node{

int num;

char name[20];

float score1;

float score2;

float score3;

float sum;

struct node\* next;

}stu,\*pstu;

void tailIn(pstu\* pphead, pstu\* pptail, stu s)

{

float sum;

pstu pcur, ppre, pnew;

pnew = (pstu)malloc(sizeof(stu));

memset(pnew, 0, sizeof(stu));

strcpy(pnew->name, s.name);

pnew->num = s.num;

pnew->score1 = s.score1;

pnew->score2 = s.score2;

pnew->score3 = s.score3;

sum = s.score1 + s.score2 + s.score3;

pnew->sum = sum;

pcur = \*pphead;

ppre = \*pphead;

if (NULL == pcur) {

\*pphead = pnew;

\*pptail = pnew;

}

else if (sum < pcur->sum) {

pnew->next = pcur;

\*pphead = pnew;

}

else {

while (pcur != NULL) {

if (pcur->sum > sum) {

ppre->next = pnew;

pnew->next = pcur;

break;

}

ppre = pcur;

pcur = pcur->next;

}

if (NULL == pcur)

{

(\*pptail)->next = pnew;

\*pptail = pnew;

}

}

}

int main() {

stu s = { 10001,"xiaoxiao",95.03,98.99,90.20 };

//stu s = { 0 };

FILE\* fp = fopen("d:\\text.txt", "r+");

int ret,retse,count=0;

pstu phead = NULL, ptail = NULL;

if (NULL == fp) {

perror("fopen");

}

while ((ret = fscanf(fp,"%d%s%f%f%f", &s.num, s.name, &s.score1, &s.score2, &s.score3))!= EOF) {

tailIn(&phead, &ptail,s);

printf("ret is: %d\n", ret);

count++;

//if (count == 4) break;

}

retse = fseek(fp, 0, SEEK\_SET);

for (int i = 0; i < count; i++) {

fprintf(fp,"%d %s %5.2f %5.2f %5.2f %5.2f\n", phead->num, phead->name, phead->score1, phead->score2, phead->score3, phead->sum);

phead = phead->next;

}

return 0;

}

