문제해결기법(13967005)

202135592 한웅재

소프트웨어

제출일: 2021. 11. 12

Q1. Lab1 (p. 16)

#define \_CRT\_SECURE\_NO\_WARNINGS// or scanf\_s

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

#include <ctype.h>

#include <stdbool.h>

struct ADDRESS {

char first[10];

char middle;

char second[10];

};

struct ADDRESS address[1000];

struct ADDRESS compressed\_address[1000];

void copy\_element(struct ADDRESS src[], struct ADDRESS dest[]) {

strcpy(dest->first, src->first);

dest->middle = src->middle;

strcpy(dest->second, src->second);

}

bool read\_file(const char \*fname,int \*count,struct ADDRESS data[]) {

FILE\* pFile;

pFile = fopen(fname, "r");

if (pFile == NULL) {

printf("cannot open the file!\n");

return false;

(\*count) = 0;

}

struct ADDRESS ex;

int i = 0;

while (fscanf(pFile,"%s %c %s",ex.first,&ex.middle,ex.second)==3) {

copy\_element(&ex, &data[i]);

i++;

(\*count)++;

}

fclose(pFile);

return true;

}

bool read\_compressed\_file(const char\* fname, int\* count, struct ADDRESS data[]) {

FILE\* pFile;

pFile = fopen(fname, "r");

if (pFile == NULL) {

printf("cannot open the file!\n");

return false;

(\*count) = 0;

}

struct ADDRESS ex;

int i = 0;

while (fscanf(pFile, "%3s%4s", ex.first,ex.second) == 2) {

copy\_element(&ex, &data[i]);

i++;

(\*count)++;

}

fclose(pFile);

return true;

}

bool write\_file(const char\* fname, int\* count) {

FILE\* pFile;

pFile = fopen(fname, "w");

if (pFile == NULL) {

printf("cannot open the file!\n");

return false;

}

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

fprintf(pFile, "%s%s", address[i].first, address[i].second);

}

fclose(pFile);

return true;

}

bool write\_compressed\_file(const char\* fname, int\* count) {

FILE\* pFile;

pFile = fopen(fname, "w");

if (pFile == NULL) {

printf("cannot open the file!\n");

return false;

}

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

fprintf(pFile, "%s - %s ", compressed\_address[i].first, compressed\_address[i].second);

}

fclose(pFile);

return true;

}

int main() {

char fname[30] = "address.txt";

char f2name[30] = "compressed.txt";

char f3name[30] = "decompressed.txt";

int count = 0;

int count\_b = 0;

read\_file(fname,&count,address);

write\_file(f2name,&count);

read\_compressed\_file(f2name,&count\_b,compressed\_address);

write\_compressed\_file(f3name, &count\_b);

return 0;

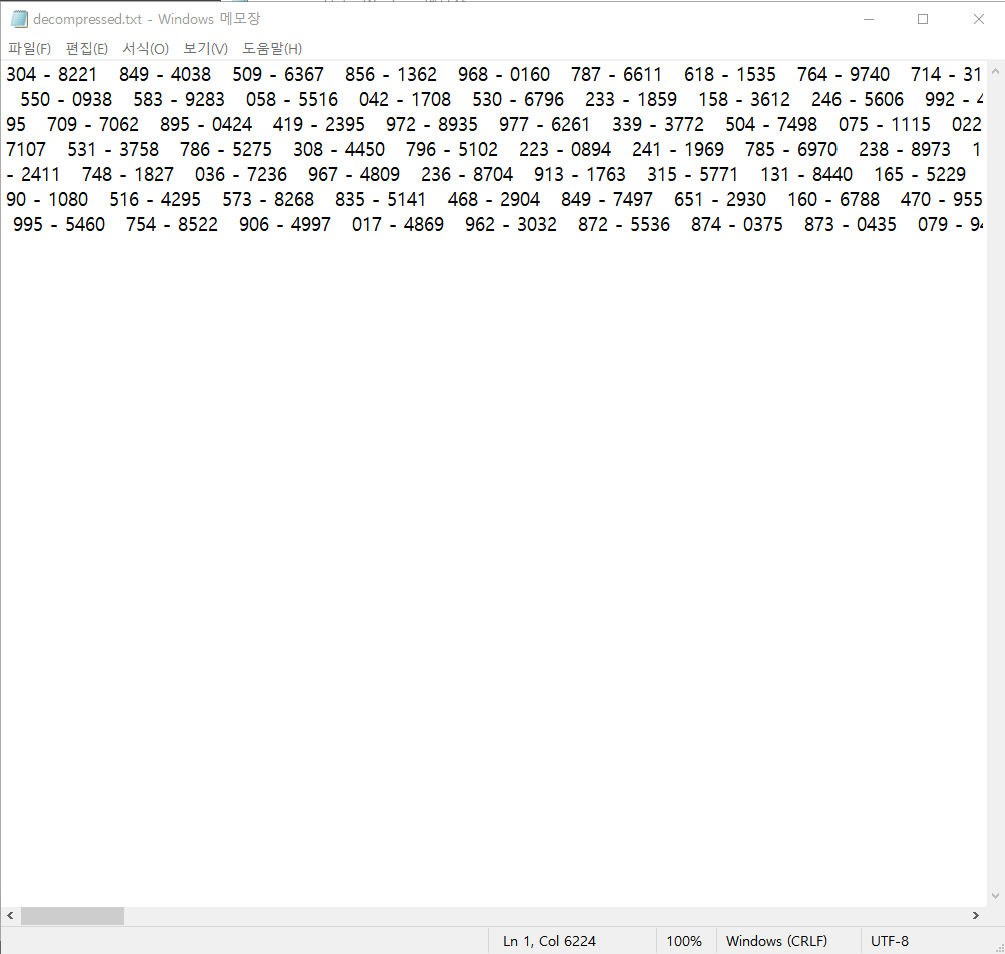
}

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

Q2. Lab2 (p. 24)

#define \_CRT\_SECURE\_NO\_WARNINGS// or scanf\_s

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

#include <ctype.h>

#include <stdbool.h>

#define KEY 3;

void do\_cypher(char\* str) {

int i = 0;

int key = KEY;

while (str[i] != '\0') {

str[i];

if (str[i] >='A' && str[i] <='Z') {

str[i] = (str[i] - 'A' + key) % 26 + 'A';

}

else if (str[i] >='a' && str[i] <='z') {

str[i] = (str[i] - 'a' + key) % 26 + 'a';

}

i++;

}

}

void do\_decrypt(char\* str) {

int i = 0;

int key = KEY;

while (str[i] != '\0') {

str[i];

int code;

if (str[i] >= 'A' && str[i] <= 'Z') {

code = (str[i] - 'A' - key);

if (code < 0) {

code += 26;

}

str[i] = code % 26 + 'A';

}

else if (str[i] >= 'a' && str[i] <= 'z') {

code = (str[i] - 'a' - key);

if (code < 0) {

code += 26;

}

str[i] = code % 26 + 'a';

}

i++;

}

}

bool write\_file(const char\* fname,char data[][200],int \*count) {

FILE\* pFile;

pFile = fopen(fname, "w");

if (pFile == NULL) {

printf("cannot open the file!\n");

return false;

}

//write char

char cypher[100];

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

strcpy(cypher,data[i]);

do\_cypher(cypher);

fprintf(pFile,"%s",cypher);

}

//

fclose(pFile);

return true;

}

bool write\_decrypted\_file(const char\* fname, char data[][200], int\* count) {

FILE\* pFile;

pFile = fopen(fname, "w");

if (pFile == NULL) {

printf("cannot open the file!\n");

return false;

}

//write char

char decrypted[100];

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

strcpy(decrypted, data[i]);

do\_decrypt(decrypted);

fprintf(pFile, "%s", decrypted);

}

//

fclose(pFile);

return true;

}

bool read\_file(const char\* fname, char data[][200],int \*count) {

FILE\* pFile;

pFile = fopen(fname, "r");

if (pFile == NULL) {

printf("cannot open the file!\n");

\*count = 0;

return false;

}

int i = 0;

char a[200];

//read char

while(fgets(a,100,pFile)!=NULL)

{

strcpy(data[i],a);

i++;

(\*count)++;

}

//

fclose(pFile);

return true;

}

int main() {

char fname[] = "original.txt";

char f2name[] = "cypher.txt";

char f3name[] = "decrypted.txt";

int count = 0;

int count\_b = 0;

char original[11][200];

char cypher[11][200];

read\_file(fname, original, &count);

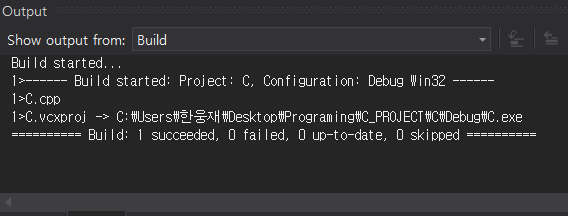
write\_file(f2name, original, &count);

read\_file(f2name,cypher,&count\_b);

write\_decrypted\_file(f3name,cypher,&count\_b);

return 0;

}

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명