**מעבדה מספר 4 במת"מ –קבצים:**

תרגיל מספר 1:

**מגישים**: **עאדל מוחסן** 207282542 **ואלן ציפין** 313206062

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdlib.h>

#include<stdio.h>

#include<string.h>

#define MAX 256

typedef struct

{

char code[10];

char\* name;

}Book;

typedef struct

{

char name[MAX];

int num\_books;

Book\* arr;

}Library;

void Get\_Lost(char\* str);

/\*

Function name:Get\_Lost

Input:char\*

Output:void

Function Algorithm: prints a string and exits the program

\*/

void input\_book(Book\* B, FILE\* in);

/\*

Function name: input\_book

Input: book\*, file\*

Output: void

Function Algorithm: reads from a txt file a book format

\*/

void input\_library(Library\* L, FILE\* in);

/\*

Function name:input\_library

Input:library\* , file\*

Output:void

Function Algorithm: reads from a txt file a library format

\*/

void output\_book(Book\* B, FILE\* out);

/\*

Function name:output\_book

Input:book\*,file\*

Output:void

Function Algorithm: write a book format to a txt file

\*/

void output\_library(Library\* L, FILE\* out);

/\*

Function name: output\_library

Input: library\*, file \*

Output:void

Function Algorithm: write a library format to a txt file

\*/

int main()

{

FILE\* in, \* out;

Library Libr;

int i;

if (!(in= fopen("input.txt", "rt")))

Get\_Lost("The input file is wrong");

input\_library(&Libr, in);

fclose(in);

if (!(out = fopen("output.txt", "wt")))

Get\_Lost("The output file is wrong");

output\_library(&Libr, out);

fclose(out);

for (i = 0; i < Libr.num\_books; i++) //free each book name that was dynamic allocated

free(Libr.arr[i].name);

free(Libr.arr); //free the whole book array in the library structure.

return 0;

}

void Get\_Lost(char\* str)

{

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

exit(1);

}

void input\_book(Book\* B, FILE\* in) {

char tempname[MAX]; //this is a temp string to save the name to get the length to the dynamic allocation

fscanf(in, "%s %s", B->code,tempname);

B->name = (char\*)malloc((strlen(tempname) + 1) \* sizeof(char)); //memo allocate string in the structure name field

if (B->name == NULL) {

Get\_Lost("Memo allocate failed");

}

strcpy(B->name,tempname); //copy the name to the name field in the book structure.

}

void input\_library(Library\* L, FILE\* in) {

int i;

fscanf(in, "%s %d", L->name, &L->num\_books);

L->arr= (Book\*)malloc((L->num\_books) \* sizeof(Book)); //memo allocate Book array in the structure book arr field

if (L->arr == NULL) {

Get\_Lost("Memo allocate failed");

}

for (i = 0; i < L->num\_books; i++) { //reads each book to the library area

input\_book(&L->arr[i], in);

}

}

void output\_book(Book\* B, FILE\* out) {

fprintf(out, "%-10s %s\n", B->code, B->name); //%-10s for correct formatting

}

void output\_library(Library\* L, FILE\* out) {

int i;

fprintf(out, "%s\n", L->name);

for (i = 0; i < L->num\_books; i++){

output\_book(&L->arr[i], out); //call output\_book to write each book from the library to the txt.

}

}

תרגיל מס' 2:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct {

char course[4];

char ID[5];

char name[17];

}Student;

void Get\_Lost(char\*);

/\*

Function name:Get\_Lost

Input:char\*

Output:void

Function Algorithm:prints a string and exits the program;

\*/

void Find(FILE\*, char\* course);

/\*

Function name:Find

Input:file\*,Char\*

Output:void

Function Algorithm: find students from same course.

\*/

int main()

{

char course[4];

FILE\* file;

file = fopen("Stud.txt", "rt");

if (file == NULL)

Get\_Lost("Cannot open file");

printf("\nEnter the name of course,up to 3 letters:");

scanf("%s", course);

Find(file, course);

fclose(file);

return 0;

}

void Get\_Lost(char\* s)

{

puts(s);

exit(1);

}

void Find(FILE\* f, char\* course)

{

char binary[17], filename[8]; //binary to save the binary number. filename to make a new file name according to the course name.

FILE\* out;

Student temp;

strcpy(filename, course); //we make a new string that combined with the course name and .txt

strcat(filename, ".txt\0");

out = fopen(filename, "wt"); //we open a new text to write only.

if (out==NULL)

Get\_Lost("Cannot open output file");

while (!feof(f)) //while we didnt reach EOF

{

fgets(temp.course, 4, f); //gets the COURSE name

fgets(temp.ID, 5, f); //gets the ID

fgets(temp.name, 17, f); //gets the STUDENT name

if (!strcmp(temp.course, course))

{

int num;

num = atoi(temp.ID); // converts the string to a integer

\_itoa(num, binary, 2); //converts the integer number to a binary number

fprintf(out, "%s, %s, %s\n", binary, temp.name, temp.course); //writes to the txt

}

}

fclose(out);

}