Катедра “Програмиране и компютърни технологии”

**Задание No 10** **дисциплина: ПИК II – курсова задача**

Студент:

Да се разработи програма тип „меню” за поддържане на складово стопанство със следните изисквания:

1. **Наличностите в складовете се съхраняват в двоични файлове. За всяка стока в склада се пазят следните данни:** 
   1. • Номенклатурен номер - уникално 12 цифрено число;
   2. • Наименование на стоката - до 50 символен низ;
   3. • Единична цена - реално число;
   4. • Количество - цяло число;
   5. • Дата на производство на стоката - записана във формата ДД.ММ.ГГГГ.

**2. Да се извършват следните обработки:**

а) добавя нова стока в склада.

б) променяне на наличностите от зададена стока ( добавяне или изваждане количества от склада ). При изваждане ако заявката е за повече съществуващото количество - да не се изпълнява. Ако наличностите станат равни на 0 стоката да се изключи от склада.

в) извеждане информация за всички стоки от склада с изтекъл срок на годност (текущата дата се задава от потребителя).

г) По зададен идентификационен номер да се разпечатва информацията за доставена стока.

1. **Данните да се поддържат в динамична структура - едносвързан списък в оперативната памет на ПК.**

**ИЗИСКВАНИЯ КЪМ ОФОРМЛЕНИЕТО** Задачата да се оформи като задача, съдържаща:

• титулна страница с данни за студента, ръководителя на курсовата задача;

• текст на заданието;

• обобщен блоков алгоритъм на разработеното програмно осигуряване;

• описание на използуваните модули (функции) - прототип, входно изходни параметри и предназначение; • общо описание за функциониране на програмата (вход/изход);

• листинг на source (изходния) код на програмата;

• резултати от изпълнението на програмата (контролен пример);

• проектът да се реализира в програмната среда като проект с разделна компилация. Дата на задаване:

Преподавател: /……………………. /

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <dirent.h>

#include <string.h>

***//=============================================================***

***//INITIALIZATION PART***

***//=============================================================***

typedef struct{

char nomenclature\_number[13];

char name[51];

float price;

int amount;

char expiration\_date[11];

struct Goods\* next;

}Goods;

Goods head;

char current\_opened\_storage\_filename[30];

***//=============================================================***

***//=============================================================***

***//FUNCTIONS DECLARATIONS***

***//=============================================================***

***/\*INIT\_HEAD() - функция, която насочва началния елемент от списъка да сочи***

***към NULL, не получава параметри, защото работи с глобалния обект - head***

***\*/***void INIT\_HEAD();

***/\*Clear\_current\_opened\_storage\_filename() - функция, която изчиства името на***

***текущият отворен склад, задавайки символът на нулев иднекс да бъде равен***

***на терминиращ знак, не получава параметри, защото работи с глобален***

***масив от символи - current\_opened\_storage\_filename[30]***

***\*/***void Clear\_current\_opened\_storage\_filename();

***/\*Enter\_current\_opened\_storage\_filename() - функция, която изисква от***

***потребителя да въведе име на склад и го задава като съдържание на глобалния***

***масив - current\_opened\_storage\_filename[30], добавяйки .bin накрая на името***

***за улеснение при отваряне на бинарния файл в бъдеще***

***\*/***void Enter\_current\_opened\_storage\_filename();

***/\*Show\_existing\_storages() - функция обхождаща директорията, в която се намира***

***текущата изпълнима програма и извежда на екрана имената на всички файлове***

***с разширение .bin (имената на складовете) ако са създадени складове***

***\*/***void Show\_existing\_storages();

***/\*Create\_a\_new\_storage() - функция за създаване на нов склад. Функцията***

***изисква от потребителя да въведе име на склада и създава бинарен файл с***

***въведеното от потребителя име (функцията проверява за дублиране и пита***

***потребителя дали иска да замени съществуващия файл ако има дублиране)***

***\*/***void Create\_a\_new\_storage();

***/\*Open\_a\_storage\_by\_name() - Функцията извиква по-горе описаната -***

***Enter\_current\_opened\_storage\_filename() функция и проверява дали***

***съществува такъв бинарен файл, ако не съществува извежда съобщение за***

***грешка и извиква функцията - Clear\_current\_opened\_storage\_filename()***

***\*/***void Open\_a\_storage\_by\_name();

***/\*Add\_goods\_in\_opened\_storage() - Функцията проверява дали е отворен склад,***

***ако е отворен склад извиква функцията за добавяне на стока към списъка -***

***Add\_goods(), след което запазва списъка в бинарния файл с името на отворения***

***склад извиквайки функцията - Save\_current\_list\_to\_file(). Ако не е отворен***

***склад извежда съобщение за грешка подканващо потребителя да отвори склад***

***\*/***void Add\_goods\_in\_opened\_storage();

***/\*Show\_all\_goods\_in\_opened\_storage() - Функцията проверява дали е отворен склад,***

***ако е отворен склад обхожда текущия списък започвайки от head и извежда на екрана***

***информация за всички стоки използвайки функцията - Print\_goods\_info\_by\_address(Goods\* goods)***

***\*/***void Show\_all\_goods\_in\_opened\_storage();

***/\*Change\_amount\_of\_goods\_by\_nomenclature\_number\_in\_opened\_storage() - Функцията проверява дали е отворен склад,***

***ако е отворен склад подканва потребителя да въведе номенклатурен номер, взима адреса***

***на елемента с въведения номенклатурен номер ако има такъв елемент използвайки функцията -***

***Return\_address\_by\_nomenclature\_number\_in\_opened\_storage(char\* nomenclature\_number),***

***извежда на екрана информацията за стоката с въведения номенклатурен номер използвайки -***

***Print\_goods\_info\_by\_address(Goods\* goods), подканва потребителя да въведе ново количество на стоката,***

***ако новото количество е равно на нула, се извиква функцията за изтриване на стока - Delete\_goods\_by\_address(Goods\* goods).***

***На финала текущото състояние на списъка се записва в бинарния файл чрез използване на функцията - Save\_current\_list\_to\_file()***

***\*/***void Change\_amount\_of\_goods\_by\_nomenclature\_number\_in\_opened\_storage();

***/\*Show\_all\_expired\_goods\_in\_opened\_storage() - Функцията проверява дали е отворен склад,***

***ако е отворен склад подканва потребителя да въведе текущата дата и я сравнява с датите на***

***изтичане на срока на годност на стоките. Извежда на екрана информацията за стоките с изтекъл срок***

***използвайки функцията - Print\_goods\_info\_by\_address(Goods\* goods).***

***\*/***void Show\_all\_expired\_goods\_in\_opened\_storage();

***/\*Show\_info\_for\_goods\_by\_nomenclature\_number\_in\_opened\_storage() - Функцията проверява дали е отворен склад,***

***ако е отворен склад подканва потребителя да въведе номенклатурен номер на стока, обхожда списъка***

***итерирайки през всички елементи и извършвайки сравнение на номенклатурните им номера с въведения от потребителя номер.***

***При съвпадение на номенклатурния номер информацията за стоката се извежда на екрана чрез използване на функцията -***

***Print\_goods\_info\_by\_address(Goods\* goods).***

***\*/***void Show\_info\_for\_goods\_by\_nomenclature\_number\_in\_opened\_storage();

***/\*Run\_an\_option(int option) - Функцията получава аргумент - въведена опция от потребителя***

***число от 1 до 9 и предприема съответната операция (извиква подходящата функция) използвайки switch***

***\*/***void Run\_an\_option(int option);

***/\*menu() - Функцията извежда на екрана меню и подканва потребителя да въведе опция***

***ако въведената опция е в интервала [1;9] връща това число, ако не - връща (-1)***

***\*/***int menu();

***/\*check\_if\_file\_is\_bin(char\* filename) - Функцията получава като параметър име на файл и връща информация дали той***

***е с разширение .bin под формата на 1 за "да" и 0 за "не"***

***\*/***int check\_if\_file\_is\_bin(char\* filename);

***/\*Clear\_list() - Функцията итерира през списъка освобождавайки паметта на всички елементи,***

***накрая насочва head към NULL, използвайки функцията INIT\_HEAD()***

***\*/***void Clear\_list();

***/\*Check\_if\_a\_storage\_is\_opened() - Функцията проверява дали current\_opened\_storage\_filename е празно***

***и ако е празно връща 0, ако не е празно връща 1***

***\*/***int Check\_if\_a\_storage\_is\_opened();

***/\*Add\_goods() - Функцията итерира през списъка до достигане на неговия край.***

***Подканва потребителя да въведе стойности на всяко поле за новата стока, записвайки***

***новата стока накрая на списъка***

***\*/***void Add\_goods();

***/\*Save\_current\_list\_to\_file() - Функцията отваря за писане файла с текущото име на склада***

***(current\_opened\_storage\_filename[30]), преброява елементите в текущия списък използвайки функцията***

***Count\_goods\_in\_memory(), записва бройката на елементите в началото на бинарния файл под***

***формата на int, след което записва всички обекти от списъка във файла и накрая затваря въпросния файл***

***\*/***void Save\_current\_list\_to\_file();

***/\*Print\_goods\_info\_by\_address(Goods\* goods) - Функцията приема за аргумент указател към***

***стока и извежда на екрана член-променливите на стоката т.е. поленцата с информация***

***\*/***void Print\_goods\_info\_by\_address(Goods\* goods);

***/\*Count\_goods\_in\_memory() - Функцията итерира през целия списък, като отброява елементите в него***

***и връща този брой под формата на int (цяло число)***

***\*/***int Count\_goods\_in\_memory();

***/\*Load\_goods\_from\_file() - Функцията проверява дали е отворен склад (дали името на текущия склад не е празно).***

***Ако е отворен склад тя извиква Clear\_list() с помощта, на която изпразва текущия списък. Опитва да отвори***

***бинарния файл с името на склада в режим за четене, ако успее прочита всички стоки от него и ги записва***

***последователно в списъка.***

***\*/***void Load\_goods\_from\_file();

***/\*Delete\_goods\_by\_address(Goods\* goods) - Функцията обхожда списъка търсейки съвпадение на адрес на някой от***

***елементите - с подадения към нея като аргумент адрес за изтриване на елемент. Когато намери съвпадение***

***освобождава паметта на съответния елемент и се грижи да насочи всички указатели коректно.***

***\*/***void Delete\_goods\_by\_address(Goods\* goods);

***/\*Return\_address\_by\_nomenclature\_number\_in\_opened\_storage(char\* nomenclature\_number) - Функцията обхожда списъка***

***търсейки съвпадение на подадения, като аргумент номенклатурен номер с някой от номенклатурните номера на стоките,***

***ако е намерено - функцията връща адрес към стоката със съответния номенклатурен номер.***

***\*/***Goods\* Return\_address\_by\_nomenclature\_number\_in\_opened\_storage(char\* nomenclature\_number);

***/\*Return\_reversed\_date(char\* date) - Функцията приема, като аргумент дата под формата на символен низ във формата DD.MM.YYYY,***

***заделя памет за нов символен низ, в който записва датата във формата YYYY.MM.DD и връща указателя към тази памет под формата на char\****

***\*/***char\* Return\_reversed\_date(char\* date);

**//=============================================================**

int main(){

INIT\_HEAD();

Clear\_current\_opened\_storage\_filename();

int option;

char c;

while(1){

option = menu();

if(option==9){

printf("exiting program...\nBYE! :)\n");

break;

}else{

Run\_an\_option(option);

}

printf("\n--->press ENTER to show menu again<---\n");

scanf("%c",&c);

scanf("%c",&c);

}

return 0;

}

***//=============================================================***

***//PRIMARY FUNCTIONS PART***

***//=============================================================***

void Show\_existing\_storages(){

int if\_exists\_any\_storage = 0;

char cwd[1024];

if (getcwd(cwd, sizeof(cwd)) != NULL){

//fprintf(stdout, "Current working dir: %s\n", cwd);

DIR \*dir;

struct dirent \*ent;

if ((dir = opendir (cwd)) != NULL) {

/\* print all the files and directories within directory \*/

printf("\n\n\t==========================================\n");

printf("\tNOW PRINTING STORAGES\n");

printf("\t==========================================\n");

while ((ent = readdir (dir)) != NULL) {

if(check\_if\_file\_is\_bin(ent->d\_name)){

printf ("\t %s\b\b\b\b \n", ent->d\_name);

if\_exists\_any\_storage = 1;

}

}

closedir (dir);

if(!if\_exists\_any\_storage){

printf("\t No storages found!\n\t You must create some in menu option 2\n");

}

printf("\t==========================================\n");

} else {

/\* could not open directory \*/

perror ("");

}

}

}

void Create\_a\_new\_storage(){

printf("\n\n\t==========================================\n");

printf("\tNOW CREATING A NEW STORAGE\n");

printf("\t==========================================\n");

char filename[1024];

int i;

FILE\* fp;

char c = 'y';

int number\_of\_goods = 0;

while(1){

c = 'y';

printf("\t Enter the name of the new storage: ");

scanf(" %s",filename);

i = strlen(filename);

filename[i] = '.';i++;

filename[i] = 'b';i++;

filename[i] = 'i';i++;

filename[i] = 'n';i++;

filename[i] = '\0';i++;

if((fp = fopen(filename,"r"))!=NULL){

fclose(fp);

printf("\t This storage already exists. Do you want to replace it? y/n: ");

scanf("%c",&c);

scanf("%c",&c);

}

if(c == 'y'){

printf("\t The storage \"%s\b\b\b\b\" was created\n",filename);

fp = fopen(filename,"wb");

fwrite(&number\_of\_goods,sizeof(int),1,fp);

fclose(fp);

break;

}

}

printf("\t==========================================\n");

}

void Open\_a\_storage\_by\_name(){

printf("\n\n\t==========================================\n");

printf("\tNOW OPENING A STORAGE\n");

printf("\t==========================================\n");

Enter\_current\_opened\_storage\_filename();

FILE\* fp = fopen(current\_opened\_storage\_filename,"r");

if(fp != NULL){

printf("\tStorage \"%s\b\b\b\b\" is successfuly opened\n",current\_opened\_storage\_filename);

fclose(fp);

Load\_goods\_from\_file();

}else{

printf("\tStorage \"%s\b\b\b\b\" not exists\n",current\_opened\_storage\_filename);

current\_opened\_storage\_filename[0]='\0';

}

printf("\t==========================================\n");

}

void Add\_goods\_in\_opened\_storage(){

printf("\n\n\t==========================================\n");

printf("\tNOW ADDING GOODS");

if(Check\_if\_a\_storage\_is\_opened()){

printf(" TO \"%s\b\b\b\b\" STORAGE\n",current\_opened\_storage\_filename);

}else{

printf("\n");

}

printf("\t==========================================\n");

if(!Check\_if\_a\_storage\_is\_opened()){

printf("\tFirst you have to open a storage from menu!\n");

printf("\t==========================================\n");

return;

}else{

Add\_goods();

Save\_current\_list\_to\_file();

}

printf("\t==========================================\n");

}

void Show\_all\_goods\_in\_opened\_storage(){

int if\_anything\_printed = 0;

printf("\n\n\t==========================================\n");

printf("\tNOW PRINTING GOODS");

if(Check\_if\_a\_storage\_is\_opened()){

printf(" FROM \"%s\b\b\b\b\" STORAGE\n",current\_opened\_storage\_filename);

}else{

printf("\n");

}

printf("\t==========================================\n");

if(!Check\_if\_a\_storage\_is\_opened()){

printf("\tFirst you have to open a storage from menu!\n");

printf("\t==========================================\n");

return;

}else{

Goods\* temp = &head;

while(temp->next != NULL){

temp = (Goods\*)(temp->next);

printf("\n");

Print\_goods\_info\_by\_address(temp);

if\_anything\_printed = 1;

}

}

if(!if\_anything\_printed){

printf("\t No goods in this storage.\n\t You can add some from option 4 in menu\n");

}

printf("\t==========================================\n");

}

void Change\_amount\_of\_goods\_by\_nomenclature\_number\_in\_opened\_storage(){

char nomenclature\_number[13];

Goods\* Address\_to\_goods\_for\_change;

printf("\n\n\t==========================================\n");

printf("\tNOW CHANGING AMOUNT OF GOODS BY NOMENCLATURE\n\tNUMBER FROM OPENED STORAGE\n");

printf("\t==========================================\n");

if(!Check\_if\_a\_storage\_is\_opened()){

printf("\tFirst you have to open a storage from menu!\n");

printf("\t==========================================\n");

return;

}else{

printf("\t Enter nomenclature number to change goods amount: ");

scanf(" %s",nomenclature\_number);

Address\_to\_goods\_for\_change = Return\_address\_by\_nomenclature\_number\_in\_opened\_storage(nomenclature\_number);

if(Address\_to\_goods\_for\_change!=NULL){

Print\_goods\_info\_by\_address(Address\_to\_goods\_for\_change);

printf("\t Enter new amount: ");

scanf(" %d",&(Address\_to\_goods\_for\_change->amount));

if(Address\_to\_goods\_for\_change->amount==0){

Delete\_goods\_by\_address(Address\_to\_goods\_for\_change);

Address\_to\_goods\_for\_change=NULL;

}

Save\_current\_list\_to\_file();

}else{

printf("\t No such nomenclature number!\n");

}

}

printf("\t==========================================\n");

}

void Show\_all\_expired\_goods\_in\_opened\_storage(){

char current\_date[11];

char\* reversed\_current\_date;

char\* reversed\_expiration\_date;

int if\_exists\_expired = 0;

printf("\n\n\t==========================================\n");

printf("\tNOW PRINTING EXPIRED GOODS \n\tFROM OPENED STORAGE\n");

printf("\t==========================================\n");

if(!Check\_if\_a\_storage\_is\_opened()){

printf("\tFirst you have to open a storage from menu!\n");

printf("\t==========================================\n");

return;

}else{

printf("\t Enter the date of today (for example 01.01.2017): ");

scanf(" %s",current\_date);

reversed\_current\_date = Return\_reversed\_date(current\_date);

Goods\* temp = &head;

while(temp->next != NULL){

temp = (Goods\*)(temp->next);

reversed\_expiration\_date = Return\_reversed\_date(temp->expiration\_date);

if(strcmp(reversed\_current\_date,reversed\_expiration\_date)>0){

free(reversed\_expiration\_date);

printf("\n");

Print\_goods\_info\_by\_address(temp);

if\_exists\_expired = 1;

}

}

if(!if\_exists\_expired){

printf("\t No expired goods found\n");

}

free(reversed\_current\_date);

}

printf("\t==========================================\n");

}

void Show\_info\_for\_goods\_by\_nomenclature\_number\_in\_opened\_storage(){

char Nomenclature\_number[13];

int if\_something\_found = 0;

printf("\n\n\t==========================================\n");

printf("\tNOW PRINTING INFO BY NOMENCLATURE NUMBER\n");

printf("\t==========================================\n");

if(!Check\_if\_a\_storage\_is\_opened()){

printf("\tFirst you have to open a storage from menu!\n");

printf("\t==========================================\n");

return;

}else{

printf("\t Enter nomenclature number to search: ");

scanf(" %s",Nomenclature\_number);

printf("\t ------------------------------------\n");

printf("\t now searching...\n");

printf("\t ------------------------------------\n");

Goods\* temp = &head;

while(temp->next != NULL){

temp = (Goods\*)(temp->next);

if(strcmp(Nomenclature\_number,temp->nomenclature\_number)==0){

if\_something\_found = 1;

Print\_goods\_info\_by\_address(temp);

}

}

if(!if\_something\_found){

printf("\t no such nomenclature number\n");

}

}

printf("\t==========================================\n");

}

***//=============================================================***

***//=============================================================***

***//SECONDARY FUNCTIONS PART***

***//=============================================================***

char\* Return\_reversed\_date(char\* date){

char\* reversed\_date = (char\*)malloc(sizeof(char)\*11);

reversed\_date[8] = date[0];

reversed\_date[9] = date[1];

reversed\_date[7] = date[2];

reversed\_date[5] = date[3];

reversed\_date[6] = date[4];

reversed\_date[4] = date[5];

reversed\_date[0] = date[6];

reversed\_date[1] = date[7];

reversed\_date[2] = date[8];

reversed\_date[3] = date[9];

reversed\_date[10] = '\0';

return reversed\_date;

}

Goods\* Return\_address\_by\_nomenclature\_number\_in\_opened\_storage(char\* nomenclature\_number){

int if\_something\_found = 0;

Goods\* temp = &head;

while(temp->next != NULL){

temp = (Goods\*)(temp->next);

if(strcmp(nomenclature\_number,temp->nomenclature\_number)==0){

if\_something\_found = 1;

return temp;

}

}

if(!if\_something\_found){

return NULL;

}

}

void Delete\_goods\_by\_address(Goods\* goods){

Goods\* temp = &head;

Goods\* prev;

while(temp->next != NULL){

prev = temp;

temp = (Goods\*)(temp->next);

if(temp==goods){

prev->next = temp->next;

printf("\t Deleting goods because the amount is 0...\n");

//Print\_goods\_info\_by\_address(temp);

free(temp);

}

}

}

void Load\_goods\_from\_file(){

if(!Check\_if\_a\_storage\_is\_opened()){return;}

Clear\_list();

int i = 0;

int number\_of\_goods;

Goods\* temp = &head;

FILE\* fp = fopen(current\_opened\_storage\_filename,"rb");

if(fp!=NULL){

fread(&number\_of\_goods,sizeof(int),1,fp);

//printf("read %d\n",number\_of\_goods);

while(i < number\_of\_goods){

temp->next = (Goods\*)malloc(sizeof(Goods));

temp = (Goods\*)(temp->next);

fread(temp,sizeof(Goods),1,fp);

//printf("read:\n");

//Print\_goods\_info\_by\_address(temp);

temp->next = NULL;

i++;

}

fclose(fp);

}

}

int Count\_goods\_in\_memory(){

Goods\* temp = &head;

int i = 0;

while(temp->next != NULL){

temp = (Goods\*)(temp->next);

i++;

}

return i;

}

void Clear\_list(){

Goods\* temp = &head;

Goods\* temp\_next;

int if\_null = 1;

if(head.next!=NULL){

temp = (Goods\*)(temp->next);

if\_null = 0;

}

while(temp->next != NULL){

temp\_next = (Goods\*)(temp->next);

//Print\_goods\_info\_by\_address(temp);

free(temp);

temp = temp\_next;

}

if(!if\_null){

//Print\_goods\_info\_by\_address(temp);

free(temp);

}

INIT\_HEAD();

}

void Add\_goods(){

char temp\_nom\_num[13];

Goods\* temp = &head;

while(temp->next != NULL){temp = (Goods\*)(temp->next);}

temp->next = (Goods\*)malloc(sizeof(Goods));

temp = (Goods\*)(temp->next);

temp->next = NULL;

while(1){

printf("\t Enter nomenclature number (12 digits number): ");

scanf(" %s",temp\_nom\_num);

if(Return\_address\_by\_nomenclature\_number\_in\_opened\_storage(temp\_nom\_num)!=NULL){

printf("\t Sorry this nomenclature number is already used try again!\n");

}else{

strcpy(temp->nomenclature\_number,temp\_nom\_num);

break;

}

}

printf("\t Enter name (for example: potatoes): ");

scanf(" %s",temp->name);

printf("\t Enter price (for example: 5.25): ");

scanf(" %f",&(temp->price));

printf("\t Enter amount (for example: 5): ");

scanf(" %d",&(temp->amount));

printf("\t Enter expiration date (DD.MM.YYYY): ");

scanf(" %s",temp->expiration\_date);

}

void Print\_goods\_info\_by\_address(Goods\* goods){

printf("\t Nomenclature number: %s\n",goods->nomenclature\_number);

printf("\t Name: %s\n",goods->name);

printf("\t Price: %f\n",goods->price);

printf("\t Amount: %d\n",goods->amount);

printf("\t Expiration date: %s\n",goods->expiration\_date);

}

void Save\_current\_list\_to\_file(){

FILE\* fp = fopen(current\_opened\_storage\_filename,"wb");

int number\_of\_goods = Count\_goods\_in\_memory();

if(fp!=NULL){

//printf("writing %d to file\n",number\_of\_goods);

fwrite(&number\_of\_goods,sizeof(int),1,fp);

Goods\* temp = &head;

while(temp->next != NULL){

temp = (Goods\*)(temp->next);

//printf("writing this:\n");

//Print\_goods\_info\_by\_address(temp);

fwrite(temp,sizeof(Goods),1,fp);

}

fclose(fp);

}

}

int Check\_if\_a\_storage\_is\_opened(){

if(current\_opened\_storage\_filename[0] == '\0'){

return 0;

}else{

return 1;

}

}

void INIT\_HEAD(){

head.next = NULL;

}

void Clear\_current\_opened\_storage\_filename(){

current\_opened\_storage\_filename[0] = '\0';

}

void Enter\_current\_opened\_storage\_filename(){

printf("\tPlease enter the name of a storage to open: ");

scanf("%s",current\_opened\_storage\_filename);

int i = strlen(current\_opened\_storage\_filename);

current\_opened\_storage\_filename[i] = '.';i++;

current\_opened\_storage\_filename[i] = 'b';i++;

current\_opened\_storage\_filename[i] = 'i';i++;

current\_opened\_storage\_filename[i] = 'n';i++;

current\_opened\_storage\_filename[i] = '\0';

}

int check\_if\_file\_is\_bin(char\* filename){

int if\_bin = 1;

int i = 0;

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

i++;

}

i--;

if(filename[i]!='n')

if\_bin = 0;

i--;

if(filename[i]!='i')

if\_bin = 0;

i--;

if(filename[i]!='b')

if\_bin = 0;

i--;

if(filename[i]!='.')

if\_bin = 0;

return if\_bin;

}

void Run\_an\_option(int option){

switch(option){

case 1:Show\_existing\_storages();break;

case 2:Create\_a\_new\_storage();break;

case 3:Open\_a\_storage\_by\_name();break;

case 4:Add\_goods\_in\_opened\_storage();break;

case 5:Show\_all\_goods\_in\_opened\_storage();break;

case 6:Change\_amount\_of\_goods\_by\_nomenclature\_number\_in\_opened\_storage();break;

case 7:Show\_all\_expired\_goods\_in\_opened\_storage();break;

case 8:Show\_info\_for\_goods\_by\_nomenclature\_number\_in\_opened\_storage();break;

}

}

int menu(){

printf("\n\n==========================================\n");

printf("MENU\n");

printf("==========================================\n");

printf("OPTIONS FOR STORAGES\n");

printf("------------------------------------------\n");

printf(" 1.Show existing storages\n");

printf(" 2.Create a new storage\n");

printf(" 3.Open a storage by name\n");

printf("------------------------------------------\n");

printf("OPTIONS FOR GOODS\n");

printf("------------------------------------------\n");

printf(" 4.Add goods in opened storage\n");

printf(" 5.Show all goods in opened storage\n");

printf(" 6.Change amount of goods by nomenclature number in opened storage\n");

printf(" 7.Show all expired goods in opened storage\n");

printf(" 8.Show info for goods by nomenclature number in opened storage\n");

printf("------------------------------------------\n");

printf("OTHER OPTIONS\n");

printf("------------------------------------------\n");

printf(" 9.EXIT\n");

printf("==========================================\n");

int option;

printf("ENTER AN OPTION: ");

scanf(" %d",&option);

if(!((option<=9)&&(option>=1))){

option = -1;

}

return option;

}

***//=============================================================***

***/\****

***ПРИМЕР ЗА ИЗПОЛЗВАНЕ НА ПРОГРАМАТА:***

***==========================================***

***MENU***

***==========================================***

***------------------------------------------***

***1.Show existing storages***

***2.Create a new storage***

***3.Open a storage by name***

***------------------------------------------***

***OPTIONS FOR GOODS***

***------------------------------------------***

***4.Add goods in opened storage***

***5.Show all goods in opened storage***

***6.Change amount of goods by nomenclature number in opened storage***

***7.Show all expired goods in opened storage***

***8.Show info for goods by nomenclature number in opened storage***

***------------------------------------------***

***OTHER OPTIONS***

***------------------------------------------***

***9.EXIT***

***==========================================***

***ENTER AN OPTION: 2***

***==========================================***

***NOW CREATING A NEW STORAGE***

***==========================================***

***Enter the name of the new storage: Storage1***

***The storage "Storage1" was created***

***==========================================***

***--->press ENTER to show menu again<---***

***==========================================***

***MENU***

***==========================================***

***OPTIONS FOR STORAGES***

***------------------------------------------***

***1.Show existing storages***

***2.Create a new storage***

***3.Open a storage by name***

***------------------------------------------***

***OPTIONS FOR GOODS***

***------------------------------------------***

***4.Add goods in opened storage***

***5.Show all goods in opened storage***

***6.Change amount of goods by nomenclature number in opened storage***

***7.Show all expired goods in opened storage***

***8.Show info for goods by nomenclature number in opened storage***

***------------------------------------------***

***OTHER OPTIONS***

***------------------------------------------***

***9.EXIT***

***==========================================***

***ENTER AN OPTION: 3***

***==========================================***

***NOW OPENING A STORAGE***

***==========================================***

***Please enter the name of a storage to open: Storage1***

***Storage "Storage1" is successfuly opened***

***==========================================***

***--->press ENTER to show menu again<---***

***==========================================***

***MENU***

***==========================================***

***OPTIONS FOR STORAGES***

***------------------------------------------***

***1.Show existing storages***

***2.Create a new storage***

***3.Open a storage by name***

***------------------------------------------***

***OPTIONS FOR GOODS***

***------------------------------------------***

***4.Add goods in opened storage***

***5.Show all goods in opened storage***

***6.Change amount of goods by nomenclature number in opened storage***

***7.Show all expired goods in opened storage***

***8.Show info for goods by nomenclature number in opened storage***

***------------------------------------------***

***OTHER OPTIONS***

***------------------------------------------***

***9.EXIT***

***==========================================***

***ENTER AN OPTION: 4***

***==========================================***

***NOW ADDING GOODS TO "Storage1" STORAGE***

***==========================================***

***Enter nomenclature number (12 digits number): 123456789098***

***Enter name (for example: potatoes): Bananas***

***Enter price (for example: 5.25): 2.30***

***Enter amount (for example: 5): 30***

***Enter expiration date (DD.MM.YYYY): 18.05.2017***

***==========================================***

***--->press ENTER to show menu again<---***

***==========================================***

***MENU***

***==========================================***

***OPTIONS FOR STORAGES***

***------------------------------------------***

***1.Show existing storages***

***2.Create a new storage***

***3.Open a storage by name***

***------------------------------------------***

***OPTIONS FOR GOODS***

***------------------------------------------***

***4.Add goods in opened storage***

***5.Show all goods in opened storage***

***6.Change amount of goods by nomenclature number in opened storage***

***7.Show all expired goods in opened storage***

***8.Show info for goods by nomenclature number in opened storage***

***------------------------------------------***

***OTHER OPTIONS***

***------------------------------------------***

***9.EXIT***

***==========================================***

***ENTER AN OPTION: 4***

***==========================================***

***NOW ADDING GOODS TO "Storage1" STORAGE***

***==========================================***

***Enter nomenclature number (12 digits number): 234546574382***

***Enter name (for example: potatoes): Tomatoes***

***Enter price (for example: 5.25): 3.45***

***Enter amount (for example: 5): 48***

***Enter expiration date (DD.MM.YYYY): 20.05.2017***

***==========================================***

***--->press ENTER to show menu again<---***

***==========================================***

***MENU***

***==========================================***

***OPTIONS FOR STORAGES***

***------------------------------------------***

***1.Show existing storages***

***2.Create a new storage***

***3.Open a storage by name***

***------------------------------------------***

***OPTIONS FOR GOODS***

***------------------------------------------***

***4.Add goods in opened storage***

***5.Show all goods in opened storage***

***6.Change amount of goods by nomenclature number in opened storage***

***7.Show all expired goods in opened storage***

***8.Show info for goods by nomenclature number in opened storage***

***------------------------------------------***

***OTHER OPTIONS***

***------------------------------------------***

***9.EXIT***

***==========================================***

***ENTER AN OPTION: 4***

***==========================================***

***NOW ADDING GOODS TO "Storage1" STORAGE***

***==========================================***

***Enter nomenclature number (12 digits number): 984537543212***

***Enter name (for example: potatoes): Peaches***

***Enter price (for example: 5.25): 3.80***

***Enter amount (for example: 5): 12***

***Enter expiration date (DD.MM.YYYY): 12.05.2017***

***==========================================***

***--->press ENTER to show menu again<---***

***==========================================***

***MENU***

***==========================================***

***OPTIONS FOR STORAGES***

***------------------------------------------***

***1.Show existing storages***

***2.Create a new storage***

***3.Open a storage by name***

***------------------------------------------***

***OPTIONS FOR GOODS***

***------------------------------------------***

***4.Add goods in opened storage***

***5.Show all goods in opened storage***

***6.Change amount of goods by nomenclature number in opened storage***

***7.Show all expired goods in opened storage***

***8.Show info for goods by nomenclature number in opened storage***

***------------------------------------------***

***OTHER OPTIONS***

***------------------------------------------***

***9.EXIT***

***==========================================***

***ENTER AN OPTION: 5***

***==========================================***

***NOW PRINTING GOODS FROM "Storage1" STORAGE***

***==========================================***

***Nomenclature number: 123456789098***

***Name: Bananas***

***Price: 2.300000***

***Amount: 30***

***Expiration date: 18.05.2017***

***Nomenclature number: 234546574382***

***Name: Tomatoes***

***Price: 3.450000***

***Amount: 48***

***Expiration date: 20.05.2017***

***Nomenclature number: 984537543212***

***Name: Peaches***

***Price: 3.800000***

***Amount: 12***

***Expiration date: 12.05.2017***

***==========================================***

***--->press ENTER to show menu again<---***

***==========================================***

***MENU***

***==========================================***

***OPTIONS FOR STORAGES***

***------------------------------------------***

***1.Show existing storages***

***2.Create a new storage***

***3.Open a storage by name***

***------------------------------------------***

***OPTIONS FOR GOODS***

***------------------------------------------***

***4.Add goods in opened storage***

***5.Show all goods in opened storage***

***6.Change amount of goods by nomenclature number in opened storage***

***7.Show all expired goods in opened storage***

***8.Show info for goods by nomenclature number in opened storage***

***------------------------------------------***

***OTHER OPTIONS***

***------------------------------------------***

***9.EXIT***

***==========================================***

***ENTER AN OPTION: 7***

***==========================================***

***NOW PRINTING EXPIRED GOODS***

***FROM OPENED STORAGE***

***==========================================***

***Enter the date of today (for example 01.01.2017): 19.05.2017***

***Nomenclature number: 123456789098***

***Name: Bananas***

***Price: 2.300000***

***Amount: 30***

***Expiration date: 18.05.2017***

***Nomenclature number: 984537543212***

***Name: Peaches***

***Price: 3.800000***

***Amount: 12***

***Expiration date: 12.05.2017***

***==========================================***

***--->press ENTER to show menu again<---***

***==========================================***

***MENU***

***==========================================***

***OPTIONS FOR STORAGES***

***------------------------------------------***

***1.Show existing storages***

***2.Create a new storage***

***3.Open a storage by name***

***------------------------------------------***

***OPTIONS FOR GOODS***

***------------------------------------------***

***4.Add goods in opened storage***

***5.Show all goods in opened storage***

***6.Change amount of goods by nomenclature number in opened storage***

***7.Show all expired goods in opened storage***

***8.Show info for goods by nomenclature number in opened storage***

***------------------------------------------***

***OTHER OPTIONS***

***------------------------------------------***

***9.EXIT***

***==========================================***

***ENTER AN OPTION:***

***\*/***