#include <conio.h>

#include <stdio.h >

#include <stdlib.h>

#include <math.h>

#include <locale.h>

#include <string.h>

#define computer struct \_computer

#pragma

int print();

int input();

int add();

int delete();

int change();

int sort();

int sort\_int();

int save();

int restore();

computer{

char manufacture[25]; /\* Фирма-изготовитель \*/

char cpu\_type[15]; /\* Тип процессора \*/

float clock\_frequency; /\* Тактовая частота \*/

int memory; /\* Ёмкость ОЗУ \*/

int hdd\_capacity; /\* Ёмкость жёсткого диска \*/

computer\* next; /\* Указатель на следующий элемент \*/

};

computer\* computer\_fp = NULL; /\* Начало списка \*/

char fname[] = "info.DAT"; /\* Файл для хранения списка \*/

void main() {

char eoj; /\* Флаг окончания работы \*/

setlocale(LC\_ALL, "Rus");

restore();

int key = 1;

while (key != 0) {

printf("1. Вывод\n");

printf("2. Ввод\n");

printf("3. Добавить\n");

printf("4. Удалить\n");

printf("5. Изменить\n");

printf("6. Сортировать\n");

printf("7. Числовая сортировка \n");

printf("8. Сохранить\n");

printf("9. Восстановить\n");

printf("0. Выйти\n");

printf("Выберите пункт меню:");

scanf("%d%", &key);

switch (key)

{

case 1: {

print();

break;

}

case 2: {

input();

break;

}

case 3: {

add();

break;

}

case 4: {

delete();

break;

}

case 5: {

change();

break;

}

case 6: {

sort();

break;

}

case 7: {

sort\_int();

break;

}

case 8: {

save();

break;

}

case 9: {

restore();

break;

}

default:

break;

}

}

}

int print() {

computer\* a;

int i, j;

if (computer\_fp == NULL) printf("Список пуст\n");

else {

for (a = computer\_fp, i = 1, j = 1; a != NULL; a = a->next, j++, i++) {

printf("#%-2d %-10s %-3s %.2f %-2d %-2d \n",

i, a->manufacture, a->cpu\_type, a->clock\_frequency, a->memory, a->hdd\_capacity);

if (j == 20) {

printf("Нажмите любую кнопку чтобы продолжить...\n");

\_getch();

j = 1;

}

}

printf("\_\_Конец списка\_\_\n");

}

return 0;

}

int input() {

int cc;

printf("Введите имя= \* - чтобы закончить ввод\n");

/\* Конец ввода - при вводе '\*' вместо имени \*/

while (!(cc = add())); /\* Вызов функции добавления \*/

return cc;

}

int add() {

computer\* a = NULL, \*b = NULL;

char ss[40];

int i = 1;

/\* Если список существует,осуществляем вставку элемента \*/

if (computer\_fp != NULL)

for (i++, a = computer\_fp; a->next != NULL; a = a->next, i++);

/\* Приглашение к вводу \*/

printf("Строка #%d. Ввод: Изготовитель, тип процессора, тактовая частота(2,23), объём ОЗУ, объём жесткого диска\n", i);

scanf("%s", ss);

if (ss[0] == '\*') return 2;

/\* Выделение памяти под новый элемент \*/

b = (computer\*)malloc(sizeof(computer));

strcpy(b->manufacture, ss);

scanf("%s %f %d %d", &(b->cpu\_type), &(b->clock\_frequency), &(b->memory), &(b->hdd\_capacity));

b->next = NULL;

if (computer\_fp == NULL) computer\_fp = b;

else a->next = b;

return 0;

}

int save() {

FILE\* dat;

computer\* a;

dat = fopen(fname, "w"); /\* Открытие файла на запись \*/

/\* Запись в файл осуществляется полями \*/

for (a = computer\_fp; a != NULL; a = a->next)

fprintf(dat, "%s %s %f %d %d\n", a->manufacture, a->cpu\_type, a->clock\_frequency, a->memory, a->hdd\_capacity);

/\* В конце файла - спецкод '\*\*\*' \*/

fprintf(dat, "\*\*\*\n");

fclose(dat); /\* Закрытие файла \*/

return 0;

}

int restore() {

FILE\* dat;

char ss[40];

computer\* a = NULL, \*b = NULL;

/\* Открытие файла для чтения,если файл не найден-вывод

соответствующего сообщения \*/

if ((dat = fopen(fname, "r")) == NULL) {

printf("Файл не найден: %s\n", fname);

return 1;

}

else {

computer\_fp = NULL;

do {

/\* Чтение из файла по полям пока не дошли до

спецкода '\* '\*/

fscanf(dat, "%s", ss);

if (ss[0] != '\*') {

/\* Выделение памяти под новый элемент \*/

b = (computer\*)malloc(sizeof(computer));

if (computer\_fp == NULL) computer\_fp = b;

else a->next = b;

strcpy(b->manufacture, ss);

fscanf(dat, "%s %f %d %d\n", &(b->cpu\_type), &(b->clock\_frequency), &(b->memory), &(b->hdd\_capacity));

b->next = NULL;

a = b;

}

} while (ss[0] != '\*');

fclose(dat); /\* Закрытие файла \*/

}

return 0;

}

int sort() {

int n;

computer\* a = NULL, \*b = NULL, \*c = NULL;

/\* Если список пустой или в нем один элемент,

то выход из функции \*/

if ((computer\_fp == NULL) || (computer\_fp->next == NULL)) return 0;

/\* Сортировка списка методом "пузырька" \*/

for (n = 1; n; ) {

n = 0;

for (a = computer\_fp, b = computer\_fp->next; b != NULL; )

if (strcmp(a->manufacture, b->manufacture) > 0) {

a->next = b->next; b->next = a;

if (a == computer\_fp) computer\_fp = b;

else c->next = b;

c = b; b = a->next;

n = 1;

}

else {

c = a; a = b; b = b->next;

}

}

return 0;

}

int sort\_int() {

int n;

computer\* a = NULL, \*b = NULL, \*c = NULL;

/\* Если список пустой или в нем один элемент,

то выход из функции \*/

if ((computer\_fp == NULL) || (computer\_fp->next == NULL)) return 0;

/\* Сортировка списка методом "пузырька" \*/

for (n = 1; n; ) {

n = 0;

for (a = computer\_fp, b = computer\_fp->next; b != NULL; )

if (a->clock\_frequency > b->clock\_frequency) {

a->next = b->next; b->next = a;

if (a == computer\_fp) computer\_fp = b;

else c->next = b;

c = b; b = a->next;

n = 1;

}

else {

c = a; a = b; b = b->next;

}

}

return 0;

}

int get\_ln() {

int ln;

printf("Введите номер строки: ");

do {

scanf("%d", &ln);

if (ln < 1) {

printf("Неверный номер строки, попробуйте ещё раз:");

ln = 0;

}

} while (!ln);

return ln;

}

int delete() {

int ln;

computer\* a, \*b;

/\* Если списка нет в памяти,то вывод соотвтствуюшего

сообщения \*/

if (computer\_fp == NULL) printf("List empty\n");

/\* Иначе-ввод номера элемента с помощью функции GET\_LN \*/

else {

ln = get\_ln() - 1;

if (!ln) {

/\* Если номер корректен - переприсваивание указателей

и освобождение памяти \*/

a = computer\_fp; computer\_fp = a->next; free(a);

}

else {

for (ln--, a = computer\_fp; ln && (a != NULL); a = a->next, ln--);

if (a != NULL)

if ((b = a->next) != NULL) {

a->next = b->next; free(b);

}

}

}

return 0;

}

int change() {

char ss[40];

int ln;

computer\* a;

ln = get\_ln() - 1; /\* Ввод номера элемента \*/

for (a = computer\_fp; ln && a != NULL; ln--, a = a->next);

if (ln) return 0;

/\* Вывод старых и ввод новых значений \*/

/\* Запись новых значений в список \*/

printf("Изготовитель = %s Новый изготовитель = ", a->manufacture);

gets\_s(ss, 40);

gets\_s(ss, 40);

if (\*ss) strcpy(a->manufacture, ss);

printf("Тип процессора = %s Новый тип процессора = ", a->cpu\_type);

gets\_s(ss, 40);

if (\*ss) sscanf(ss, "%s", &(a->cpu\_type));

printf("Тактовая частота = %f Новый тактовая частота = ", a->clock\_frequency);

gets\_s(ss, 40);

if (\*ss) sscanf(ss, "%f", &(a->clock\_frequency));

printf("Объём ОЗУ = %d Новый объём ОЗУ = ", a->memory);

gets\_s(ss, 40);

if (\*ss) sscanf(ss, "%d", &(a->memory));

printf("Объём жёсткого диска = %d Новый объём жёсткого диска = ", a->hdd\_capacity);

gets\_s(ss, 40);

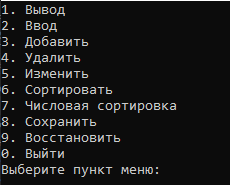
if (\*ss) sscanf(ss, "%d", &(a->hdd\_capacity));

return 0;

}

**Результат работы программы**

Меню:



Ввод:



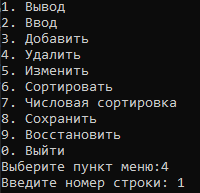
Вывод:  
  
  
Сортировка по алфавиту (сортируется по полю «Изготовитель»):



Числовая сортировка (сортируется по полю «Тактовая частота»):



Удаление:





Восстановление:



Изменение:

