**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего профессионального образования   
"Национальный исследовательский университет   
"Высшая школа экономики"**

Московский институт электроники и математики Национального

исследовательского университета "Высшая школа экономики"

Департамент прикладной математики

**ОТЧЕТ**

**По лабораторной работе №8**

**Структуры**

**По курсу «Алгоритмизация и программирование»**

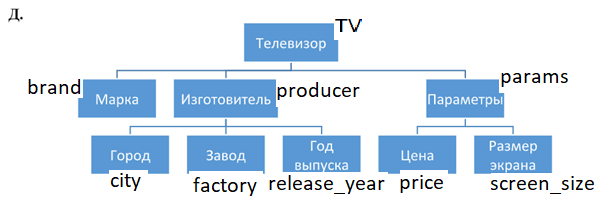
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | ФИО студента | | Номер группы | Дата |
| Кармаев Александр Андреевич | БПМ211 | 15.10.2021 |
|  |
|  |
|  |

**Москва – 2021 г.**

**ЗАДАНИЕ (вариант №13)**

*Изображение выглядит как текст

Автоматически созданное описание*

****

****

**РЕШЕНИЕ**

**header\_8.h**

struct producer { // структура Производитель

char city[64]; // строка Города (Например - "Tver")

char factory[64]; // строка Завода (Например - "Jabil Circuit")

int release\_year; // Год выпуска (Например - 2009

};

struct params { // структура Параметры

int price; // Цена (Например - 255000) - в рублях

int screen\_size; // Размер экрана (Например - 65) - в дюймах

};

struct TV { // структура Телевизор

char brand[64]; // строка Марки (Например - "Sony")

struct producer Produser; // экземпляр структуры Производитель

struct params Params; // экземпляр структуры Параметры

};

// { "brand", { "city", "factory", release\_year }, { price, screen\_size } }

**main.c**

#include <stdio.h>

#include <string.h> // библиотека для работы со строками

#include <stdlib.h> // библиотека для работы с памятью для создания массива, неизвестного размера(динамическое выделение памяти)

#include "header\_8.h" // подключаем в программу заголовочный файл с нашими структурами

#define CHOISE 1 // способ ввода данных в массив пользователем (может быть 2, тогда будет выбран способ решения задачи через заранее созданный массив)

void print\_TV\_array(struct TV\* arr, int n) { // функция вывода элементов массива

printf("\nTelevisions:\n");

for (int i = 0; i < n; i++)

printf("%d -> Brand - %s, City - %s, Factory - %s, Release year - %d, Price - %d rubles, Screen size - %d'' \n", i + 1, arr[i].brand, arr[i].Produser.city, arr[i].Produser.factory, arr[i].Produser.release\_year, arr[i].Params.price, arr[i].Params.screen\_size);

printf("\n");

}

void number\_input(char str[256], int\* p) { // функция присвоит необходимое(числовое) значение переменной в основной функции, находящейся по адресу(2-ой аргумент)

char string[256];

printf("%s", str);

fgets(string, sizeof(string), stdin);

while (sscanf(string, "%i", p) != 1) {

printf("Incorrect input, try again!! %s", str);

fgets(string, sizeof(string), stdin);

}

}

void input\_in\_array(struct TV \*arr, int n) { // функция ввода данных в массив структур

struct TV Arr[] = {

{ "Sony", { "Shenzhen", "Foxconn", 2010 }, { 255000, 65 } }, { "LG", { "Shenzhen", "Foxconn", 2010 }, { 175000, 55 } }, { "Samsung", { "Shenzhen", "Foxconn", 2010 }, { 21280, 32 } }, { "Panasonic", { "Shenzhen", "Foxconn", 2010 }, { 35000, 48 } },

{ "Sony", { "Shenzhen", "Foxconn", 2011 }, { 255000, 65 } }, { "LG", { "Shenzhen", "Foxconn", 2011 }, { 175000, 55 } }, { "Samsung", { "Shenzhen", "Foxconn", 2011 }, { 21280, 32 } }, { "Panasonic", { "Shenzhen", "Foxconn", 2011 }, { 35000, 48 } },

{ "Sony", { "Shenzhen", "Foxconn", 2012 }, { 255000, 65 } }, { "LG", { "Shenzhen", "Foxconn", 2012 }, { 175000, 55 } }, { "Samsung", { "Shenzhen", "Foxconn", 2012 }, { 21280, 32 } }, { "Panasonic", { "Shenzhen", "Foxconn", 2012 }, { 35000, 48 } },

{ "Sony", { "Shenzhen", "Foxconn", 2013 }, { 255000, 65 } }, { "LG", { "Shenzhen", "Foxconn", 2013 }, { 175000, 55 } }, { "Samsung", { "Shenzhen", "Foxconn", 2013 }, { 21280, 32 } }, { "Panasonic", { "Shenzhen", "Foxconn", 2013 }, { 35000, 48 } },

{ "Sony", { "Shenzhen", "Foxconn", 2014 }, { 255000, 65 } }, { "LG", { "Shenzhen", "Foxconn", 2014 }, { 175000, 55 } }, { "Samsung", { "Shenzhen", "Foxconn", 2014 }, { 21280, 32 } }, { "Panasonic", { "Shenzhen", "Foxconn", 2014 }, { 35000, 48 } }

}; // массив с заготовленными структурами, которые может выбрать пользователь, чтобы не создавать свои собственные

int size = sizeof(Arr) / sizeof(Arr[0]);

print\_TV\_array(Arr, size);

printf("Choose some of those structers & input their numbers, or write 0 and create your own (you can create only %d): \n", n);

int number = 0; // счётчик текущего индекса нашего массива структур

do { // пользователь выбирает заготовленные структуры

int i = 0;

do {

printf("Input arr[%d] [1, %d] : ", number, size+1);

number\_input("", &i);

} while (i < 0 || i > size + 1);

if (i == 0)

break;

arr[number] = Arr[i - 1];

number++;

} while (number != n);

while (number != n) { // пользователь создаёт свои собственные структуры

printf("\t create arr[%d]: \n", number);

printf("Brand - ");

scanf("%s", &(arr[number].brand));

printf("City - ");

scanf("%s", &(arr[number].Produser.city));

printf("Factory - ");

scanf("%s", &(arr[number].Produser.factory));

getchar();

number\_input("Release year - ", &(arr[number].Produser.release\_year));

number\_input("Price (in rubles) - ", &(arr[number].Params.price));

number\_input("Screen size (in inches) - ", &(arr[number].Params.screen\_size));

number++;

}

}

void print\_answer(struct TV arr[], int n, int year) { // функция, которая выведет ответ на условие задачи варианта

int max\_cnt = 0; // переменная будет содержать максимальное количество повторений одинаковых брендов структуры в массиве

int max\_ind = 0; // переменная будет содержать индекс этой структуры, бренд которой повторяется чаще всего

int\* inds = NULL; // массив будет содержать индексы структур, бренд которых повторяется чаще всего в массиве (количество повторений одинаково - так как может быть несколько самых популярных брендов)

int size = 0; // размер массива inds

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

if (arr[i].Produser.release\_year > year) {

int count = -1; // количество повторений текущего элемента // -1, т.к. при последующем проходе будет прочитан заново элемент под индексом i

for (int j = 0; j < n; j++) {

if (arr[j].Produser.release\_year > year && !strcmp(arr[i].brand, arr[j].brand)) // смторим сколько раз определённый бренд повторяется в массиве

count++;

}

if (count > max\_cnt) { // ищем максимальное количество повторений бренда

max\_cnt = count;

max\_ind = i;

size = 1;

inds = realloc(inds, size \* sizeof(int)); // добавляем индекс структуры самого популярного бренда в наш массив

inds[0] = max\_ind;

}

else if (count == max\_cnt) { // рассматриваем ситуацию с одинаковым количеством повторений разных брендов ( - несколько самых популярных)

for (int j = 0; j < size; j++) {

if (!strcmp(arr[inds[j]].brand, arr[i].brand))

goto place;

}

size++;

inds = realloc(inds, size \* sizeof(int)); // если индекс структуры бренда ещё не был добавлен в inds, то делаем это

inds[size-1] = i;

}

place:;

}

}

if (size == 0) { // печатаем ответ в зависимости от обстоятельств (нет самого популярного, т.к. год не подошёл; один бренд - самый популярный; несколько популярных)

printf("There is not the most popular city after %d year", year);

}

else if (size == 1) {

printf("The most popular city after %d year is %s", year, arr[max\_ind].brand);

}

else {

printf("The most popular cities after %d year are %s", year, arr[max\_ind].brand);

for (int i = 1; i < size; i++)

printf(", %s", arr[inds[i]].brand);

}

}

int main(void) {

int year = 2010; // год, после которого будем искать самый популярный бренд

#if CHOISE == 1 // говорим компилятору вставить в программу определённый код в зависимости от условия

int n = 5; // переменная с количеством элементов массива структур

do {

number\_input("Input number (1-20) of elements in array: ", &n);

} while (n < 1 || n > 20);

struct TV\* arr = malloc(n \* sizeof(struct TV)); // создаём динамический массив структур TV

input\_in\_array(arr, n); // вводим значения в массив

#elif CHOISE == 2

// создаём готовый массив (можно вписать любые структуры типа TV)

struct TV arr[] = { { "Sony", { "Shenzhen", "Foxconn", 2010 }, { 255000, 65 } }, { "LG", { "Shenzhen", "Foxconn", 2010 }, { 175000, 55 } }, { "Samsung", { "Shenzhen", "Foxconn", 2010 }, { 21280, 32 } }, { "Panasonic", { "Shenzhen", "Foxconn", 2010 }, { 35000, 48 } }, { "Sony", { "Shenzhen", "Foxconn", 2011 }, { 255000, 65 } } };

int n = sizeof(arr) / sizeof(arr[0]);

#endif

print\_TV\_array(arr, n); // печатаем массив структур

number\_input("Input year according to the task condition: ", &year); // вводим год

print\_answer(arr, n, year); // выводим ответ на задачу варианта

printf("\n");

return 0;

}

**ТЕСТЫ**

**// CHOISE = 2**

**Тест № 1** // обычный пример

**Изображение выглядит как текст, экран, снимок экрана, черный

Автоматически созданное описание**

**Тест № 2** // пример содержит все возможные обрабатываемые исключения**Изображение выглядит как текст

Автоматически созданное описание**

**// CHOISE = 1**

**Тест № 3 //** обычный пример, который содержит выбор пользователем заготовленных структур и создание собственных

**Изображение выглядит как текст

Автоматически созданное описание**

**Тест № 4** // пример содержит все возможные обрабатываемые исключения и несколько самых популярных бренда**Изображение выглядит как текст

Автоматически созданное описание**

**Тест № 5** // пользователь сам вводит все значенияИзображение выглядит как текст

Автоматически созданное описание