ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ №2

Студента 3 курса, группы ИСП-308

Асылбек уулу Бакыта

Вариант 2

1.1 Ввести значение 2-х целых переменных, а и b. Направить два указателя на эти переменные. С помощью указателя увеличить значение переменной а в 2 раза если иначе b уменьшить в 2 раза.

#include <stdio.h>

int lw2Task1() {  
 float a, b;  
 float \*F = &a, \*K = &b;  
 printf("Enter the value of a and b: \n");  
 scanf\_s("%f%f", &a, &b);  
 if (a > b) \*F = \*F \* 2; else \*K = \*K / 2;  
 printf("a = %f; b = %f", a, b);  
 return 0;  
}





1.2 Описать 2 указателя на вещественный тип. Выделить для них динамическую память. Ввести значения в выделенную память с клавиатуры. Увеличить в 2 раза 1-ую переменную.

#include <stdio.h>

#include <stdlib.h>

void lw2Task2() {  
 float \*F = (float \*) malloc(sizeof(float)), \*K = (float \*) malloc(sizeof(float));  
 printf("Enter two numbers: \n");  
 scanf\_s("%f%f", &\*F, K);  
 \*F = \*F \* 2;  
 printf("%f", \*F);  
 free(F);  
 free (K);  
}



1.3 Создать динамические массивы, используя указатели. Дано 2 массива х(n) и у(m). Сколько раз встречается второй элемент первого массива х(n) во втором массиве у(m).

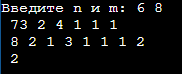
#include <stdio.h>

#include <malloc.h>

#include <time.h>

#include <stdlib.h>

int lw2Task3() {  
 srand(time(nullptr));  
 int n, m, k = 0;  
 printf("Enter the value of n and m:");  
 scanf\_s("%d%d", &n, &m);  
 if ((n <= 0) || (m <= 0)) return printf("error");  
 int\* X = (int\*) malloc(n \* sizeof(int)), \*Y = (int\*) malloc(m \* sizeof(int));  
 printf("array X: ");  
 for (int i = 0; i < n; i++) {  
 X[i] = rand() % 10;  
 printf(" %d", X[i]);  
 }  
 printf("\narray Y: ");  
 for (int i = 0; i < m; i++) {  
 Y[i] = rand() % 10;  
 printf(" %d", Y[i]);  
 }  
 for (int i = 0; i < m; i++) {  
 if (X[1] == Y[i]) k++;  
 }  
 printf("\n %d", k);  
 free(X);  
 free(Y);  
 return 0;  
}



1.4 Создать динамические массивы, используя указатели. Дан массив A(n). Переписать в массив B(n) только положительные элементы массива A, умноженные на 3. (со сжатием., без пустых элементов внутри) Затем упорядочить методом «выбора и перестановки» по возрастанию новый массив.

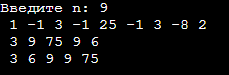
#include <stdio.h>

#include <malloc.h>

#include <time.h>

#include <stdlib.h>

int lw2Task4() {  
 srand(time(nullptr));  
 int n;  
 printf("Enter the value of n:");  
 scanf\_s("%d", &n);  
 if (n <= 0) return printf("error");  
 int \*A = (int\*) malloc(n \* sizeof(int)), m = 0, z = 1, k = 0;  
 printf("array A:");  
 for (int i = 0; i < n; i++) {  
 A[i] = z \* (rand() % 10);  
 z = -z;  
 if(A[i] > 0) k++;  
 printf(" %d", A[i]);  
 }  
 printf("\n");  
 int\* B = (int\*) malloc(k \* sizeof(int));  
 for (int i = 0; i < n; i++) {  
 if(A[i] > 0) B[m++] = A[i] \* 3;  
 }  
 printf("array B:");  
 for (int i = 0; i < m; i++) {  
 printf(" %d", B[i]);  
 }  
 printf("\n");  
 for (int i = 0; i < k - 1; i++) {  
 for (int j = i + 1; j < k; j++) {  
 if (B[i] > B[j]) {  
 z = B[j];  
 B[j] = B[i];  
 B[i] = z;  
 }  
 }  
 }  
 printf("sorted array B:");  
 for (int i = 0; i < k; i++) {  
 printf(" %d", B[i]);  
 }  
 free(A);  
 free(B);  
 return 0;  
}



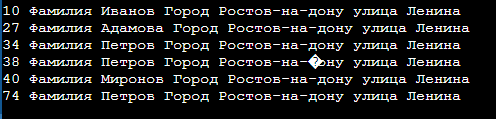
6.2 Определить комбинированный (структурный) тип для представления анкеты жителя, состоящей из его фамилии, названия города, где он проживает, и городского адреса. Адрес состоит из полей: «улица», «дом», «квартира». Ввести информацию по 100 жителям. Вывести фамилии жителей, которые живут в Ростове-на-Дону на улице Ленина.

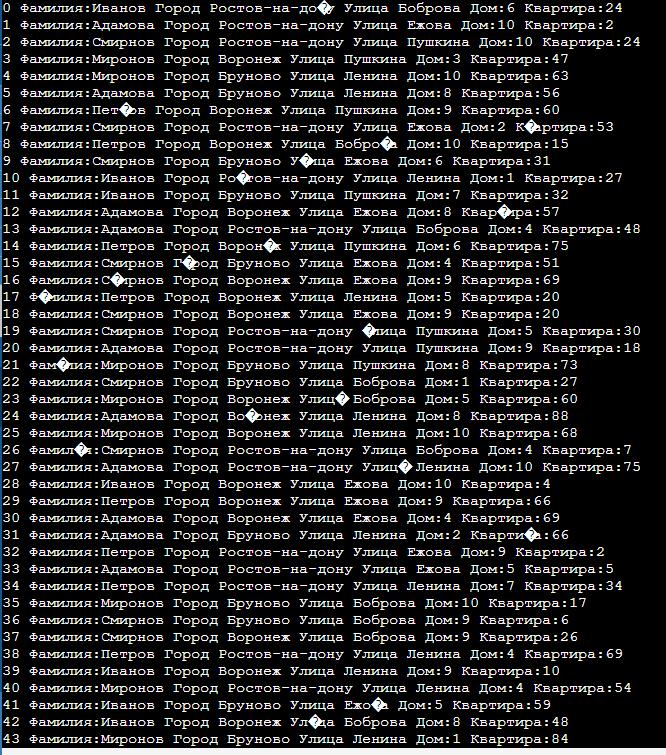
#include <stdio.h>

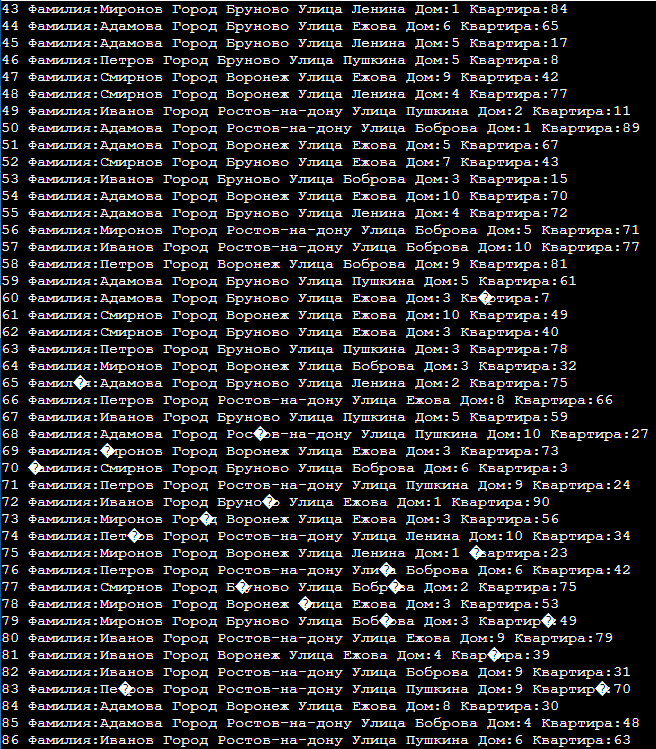
#include <stdlib.h>

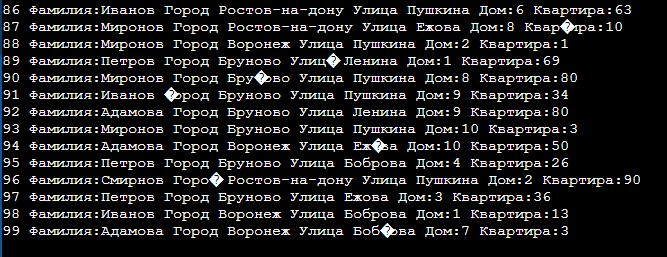
#include <string.h>

struct Address {  
 int apartment;  
 int home;  
 char\* street;  
} typedef address;  
struct Profile {  
 char\* surname;  
 char\* city;  
 address address;  
} typedef profile;  
char\* getPersonSurname(char surnames[5][10]) {  
 char \*surname = calloc(10, sizeof(char));  
 int f = rand() % 5;  
 for (int i = 0; i < 10; i++) {  
 if (surnames[f][i] == '\0') continue;  
 surname[i] = surnames[f][i];  
 }  
 return surname;  
}  
char\* getCityName(char cities[3][10]) {  
 char \*city = calloc(10, sizeof(char));  
 int f = rand() % 3;  
 for (int i = 0; i < 10; i++) {  
 if (cities[f][i] == '\0') continue;  
 city[i] = cities[f][i];  
 }  
 return city;  
}  
char\* getStreetName(char roads[4][6]) {  
 char \*street = calloc(6, sizeof(char));  
 int f = rand() % 4;  
 for (int i = 0; i < 6; i++) {  
 if (roads[f][i] == '\0') continue;  
 street[i] = roads[f][i];  
 }  
 return street;  
}  
profile\* createProfiles(int n) {  
 char surnames[5][10] = {"Adamova", "Mironov", "Smirnov", "Ivanova", "Petrova"};  
 char cities[3][10] = {"Voronezh", "rostov", "burunovo"};  
 char roads[4][6] = {"ezhova", "bobrov", "pushka", "lenina"};  
 profile \*profiles = (profile\*) malloc(n \* sizeof(profile));  
 for (int i = 0; i < n; i++) {  
 profiles[i].surname = getPersonSurname(surnames);  
 profiles[i].city = getCityName(cities);  
 profiles[i].address.street = getStreetName(roads);  
 profiles[i].address.home = rand() % 9 + 1;  
 profiles[i].address.apartment = rand() % 90 + 1;  
 }  
 return profiles;  
}  
int lw2Task5() {  
 const int n = 100;  
 srand(time(nullptr));  
 profile \*profiles = createProfiles(n);  
 printf("---------------------------------------------------------------------------------\n");  
 for (int i = 0; i < n; i++) {  
 printf("| %d \t|surname %s |city %s\t|street %s\t|home %d\t|apartment %d\t|\n", i, profiles[i].surname, profiles[i].city, profiles[i].address.street, profiles[i].address.home, profiles[i].address.apartment);  
 }  
 printf("---------------------------------------------------------------------------------\n");  
 for (int i = 0; i < n; i++) {  
 char\* str1 = strstr(profiles[i].city, "rostov");  
 char\* str2 = strstr(profiles[i].address.street, "lenina");  
 if (str1 != **NULL** && str2 != **NULL**) {  
 printf("| %d \t|surname %s |city %s |street %s|\n", i, profiles[i].surname, profiles[i].city, profiles[i].address.street);  
 }  
 }  
 for (int i = 0; i < n; i++) {  
 free(profiles[i].surname);  
 free(profiles[i].city);  
 free(profiles[i].address.street);  
 }  
 free(profiles);  
 return 0;  
}









6.4 Определить структурный тип, описывающий расписание полетов самолетов (пункт назначения, время отправления, время прибытия, время полета, стоимость билета). Заполнить структурный массив 10-ю записями. Переписать из исходного массива в другой массив, информацию только о тех рейсах, пункт назначения которых содержит по 2 буквы «а». Затем новый массив отсортировать по пункту назначения по алфавиту. (рационально переставлять все поля структуры разом)

#include <stdio.h>

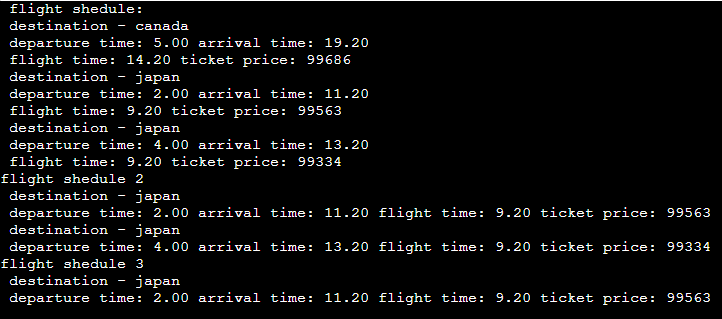
#include <math.h>

#include <time.h>

#include <stdlib.h>

#include <string.h>

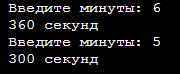
struct FlightSchedule {  
 char\* destination;  
 float departureTime;  
 float arrivalTime;  
 float flightTime;  
 int ticketPrice;  
} typedef flightSchedule;  
char\* getRandomCountry(char arrayOfDestinations[4][9]) {  
 char \*country = calloc(9, sizeof(char));  
 int f = rand() % 4;  
 for (int i = 0; i < 9; i++) {  
 if (arrayOfDestinations[f][i] == '\0') continue;  
 country[i] = arrayOfDestinations[f][i];  
 }  
 return country;  
}  
flightSchedule\* createFlightSchedule(int n, char countries[4][9]) {  
 flightSchedule\* flightSchedules = calloc(n, sizeof(flightSchedule));  
 for (int i = 0; i < n; i++) {  
 flightSchedules[i].destination = getRandomCountry(countries);  
 flightSchedules[i].departureTime = rand() % 24 + 1;  
 flightSchedules[i].arrivalTime = rand() % 24 + 1;  
 flightSchedules[i].flightTime = fabsf(flightSchedules[i].arrivalTime - flightSchedules[i].departureTime);  
 flightSchedules[i].ticketPrice = rand() % 1000 + 99000;  
 }  
 return flightSchedules;  
}  
void printFlightSchedule(flightSchedule flightSchedules[], int n) {  
 printf("--------------------------------------------------------------------------\n");  
 for (int i = 0; i < n; i++) {  
 printf("| %d ", i);  
 printf("| destination - %s\t", flightSchedules[i].destination);  
 printf("| ticket price: %d |\n", flightSchedules[i].ticketPrice);  
 printf("| %d ", i);  
 printf("| departure time: %.2f\t", flightSchedules[i].departureTime);  
 printf("| arrival time: %.2f ", flightSchedules[i].arrivalTime);  
 printf("| flight time: %.2f|\n", flightSchedules[i].flightTime);  
 }  
 printf("--------------------------------------------------------------------------\n");  
}  
void lw2Task6() {  
 const int n = 10;  
 srand(time(nullptr));  
 char countries[4][9] = {"japan", "canada", "argentina", "france"};  
 int k = 0, z = 0;  
 flightSchedule \*flightSchedules1 = createFlightSchedule(n, countries), \*flightSchedules2 = calloc(n, sizeof(flightSchedule));  
 printf(" flight schedule:\n");  
 printFlightSchedule(flightSchedules1, n);  
 for (int i = 0; i < n; i++) {  
 for (int j = 0; j < 9; j++) {  
 if (flightSchedules1[i].destination[j] == 'a') k++;  
 }  
 if (k == 2) {  
 flightSchedules2[z] = flightSchedules1[i];  
 z++;  
 }  
 k = 0;  
 }  
 printf("flight schedule 2\n");  
 printFlightSchedule(flightSchedules2, z);  
 printf("sorted flight schedule\n");  
 flightSchedule temp;  
 for (int i = 0; i < z - 1; i++) {  
 for (int j = i + 1; j < z; j++) {  
 if (strcmp(flightSchedules2[i].destination, flightSchedules2[j].destination) > 0) {  
 temp = flightSchedules2[i];  
 flightSchedules2[i] = flightSchedules2[j];  
 flightSchedules2[j] = temp;  
 }  
 }  
 }  
 printFlightSchedule(flightSchedules2, z);  
 for (int i = 0; i < n; i++) {  
 free(flightSchedules1[i].destination);  
 free(flightSchedules2[i].destination);  
 }  
 free(flightSchedules1);  
 free(flightSchedules2);  
}



7.1 Создать функцию, которая переводит время, заданное в минутах в секунды. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с обязательным оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

#include <stdio.h>

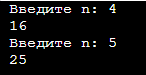
int toSec(int minute) {  
 return minute \* 60;  
}  
void lw2Task7() {  
 int minute;  
 printf("Enter the minutes:");  
 scanf\_s("%d", &minute);  
 printf(" %d seconds\n", toSec(minute));  
 printf("Enter the minutes:");  
 scanf\_s("%d", &minute);  
 printf(" %d seconds\n", toSec(minute));  
}



7.2 Реализовать функцию. Функция вычисляет площадь квадрата по заданной стороне. В главной программе задано два квадрата. Найти их площади, вызвав функцию 2 раза.

#include <stdio.h>

int areaOfSquare(int side) {  
 return side \* side;  
}  
void lw2Task8() {  
 int side;  
 printf("Enter the side of square: ");  
 scanf\_s("%d", &side);  
 printf(" %d\n", areaOfSquare(side));  
 printf("Enter the side of square: ");  
 scanf\_s("%d", &side);  
 printf(" %d\n", areaOfSquare(side));  
}

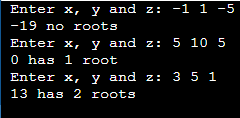


7.3 Описать функцию RootsCount(A, В, С) целого типа, определяющую количество корней квадратного уравнения (А, В, С — вещественные параметры, ). С ее помощью найти количество корней для каждого из трех квадратных уравнений с данными коэффициентами. Количество корней определять по значению дискриминанта:.

#include <stdio.h>

#include <math.h>

int rootsCount(float x, float y, float z) {  
 x = powf(y, 2) - (4 \* x \* z);  
 if(x > 0) return printf(" %.0f has 2 roots\n", x);  
 return x == 0 ? printf(" %.0f has 1 root\n", x) : printf(" %.0f no roots\n", x);  
}  
void lw2Task9() {  
 float x, y, z;  
 printf("Enter x, y and z:");  
 scanf\_s("%f%f%f", &x, &y, &z);  
 rootsCount(x, y, z);  
 printf("Enter x, y and z:");  
 scanf\_s("%f%f%f", &x, &y, &z);  
 rootsCount(x, y, z);  
 printf("Enter x, y and z:");  
 scanf\_s("%f%f%f", &x, &y, &z);  
 rootsCount(x, y, z);  
}



7.4 Описать функцию Power1(A, B) вещественного типа, находящую величину АB по формуле АВ - ехр(B\*log(A)) (параметры А и В — вещественные). В случае нулевого или отрицательного параметра А функция должна возвращать 0. С помощью этой функции в главной программе найти степени АP, ВP, CP, если даны числа Р, А, В, С.

#include <stdio.h>

#include <math.h>

float customFun(float a, float b) {  
 if (a <= 0) return 0.0f;  
 return expf(b \* logf(a));  
}  
void lw2Task10() {  
 float p, a, b, c;  
 printf("Enter values a, b, c, p:");  
 scanf\_s("%f%f%f%f", &a, &b, &c, &p);  
 printf("ap = %.1f\n", customFun(a, p));  
 printf("bp = %.1f\n", customFun(b, p));  
 printf("cp = %.1f\n", customFun(c, p));  
}

