Пензенский государственный университет

Кафедра «Вычислительная техника»

**Отчет**

По лабораторной работе №1

По курсу: «Логика и основы алгоритмизации в инженерных задачах»

на тему: «Простые структуры данных»

Выполнили:

студенты группы 20ВВ2

Макарова А. Ю.

Тельнова А. Д.

Принял:

д.т.н., профессор

Митрохин М. А.

к.т.н., доцент

Юрова О. В.

Пенза 2021

**Задание 1:** написать программу, вычисляющую разницу между максимальным и минимальным элементами массива.

**Описание работы программы:**

Объявляем переменные min, max для хранения минимального и максимального значения элемента массива. В переменную rez запишем разницу между максимальным и минимальным элементами массива.

Инициализируем массив, присваиваем переменным min и max значение первого элемента массива. В цикле сравниваем каждый элемент массива со значениями максимума и минимума. В случае если найден новый максимум или минимум, переменной max или min присваивается это значение.

for (i = 0; i < 10; i++)

{

printf("%d ", arr[i]);

if (arr[i] >= max)

max = arr[i];

if (arr[i] <= min)

min = arr[i];

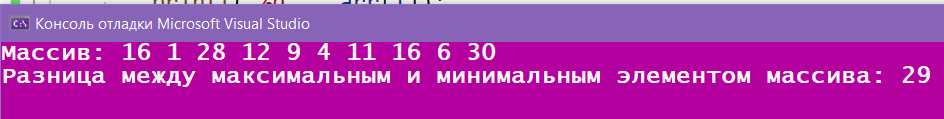
}

rez = max - min;

printf("\nРазница между максимальным и минимальным элементом массива: %d", rez);

После этого из найденного максимума вычитается минимум и выводится на экран.

**Результат работы программы:**



**Вывод:**

Написали программу, вычисляющую разницу между максимальным и минимальным элементами массива.

**Задание 2:** написать программу, реализующую инициализацию массива случайными числами.

**Описание работы программы:**

Создаем массив из 10 элементов. Инициализируем массив случайными числами с помощью функции rand, ставим ограничение на числа от 0 до 99. С помощью srand(time(NULL)) устанавливаем уникальное инициализирующее значение генератора псевдослучайных чисел, теперь при разных запусках программы каждый раз генерируются разные значения. Выводим массив на экран.

for (i = 0; i < 10; i++)

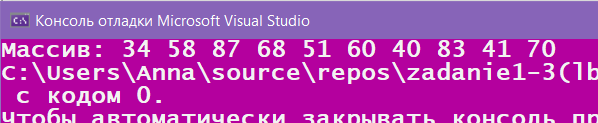
{

arr[i] = rand() % 100;

printf("%d ", arr[i]);

}

**Результат работы программы:**

****

**Вывод:**

Написали программу, реализующую инициализацию массива случайными числами с помощью функции rand.

**Задание 3:** написать программу, реализующую создание массива произвольного размера, вводимого с клавиатуры.

**Описание работы программы:**

Создаем указатель на первый элемент массива. С клавиатуры вводим размер массива, выделяем память для массива типа int размером N.

int \*arr;

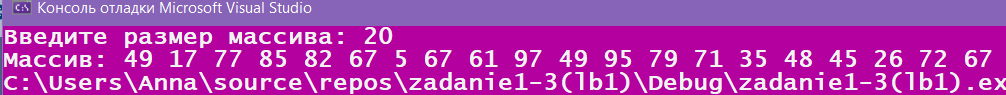
printf("Введите размер массива: ");

scanf\_s("%d", &N);

arr = (int\*)malloc(N \* sizeof(int));

Заполняем массив случайными числами, выводим на экран.

**Результат работы программы:**



**Вывод:**

Написали программу, реализующую создание массива произвольного размера, вводимого с клавиатуры, и заполнение его случайными числами.

**Задание 4:** написать программу, вычисляющую сумму значений в каждом столбце (или строке) двумерного массива.

**Описание работы программы:**

Объявляем двумерный массив из 10 строк и 10 столбцов, переменную для подсчета суммы.

Заполняем массив случайными числами и выводим на экран с указанием номеров строк и столбцов.

В цикле вычисляем сумму в каждом из столбцов: фиксируем номер столбца j, обнуляем переменную sum. Затем суммируем все элементы в столбце с индексами i от 0 до 9. Выводим полученный результат на экран под соответствующим столбцом.

printf("Cуммы:");

for (j = 0; j < 10; j++)

{

sum = 0;

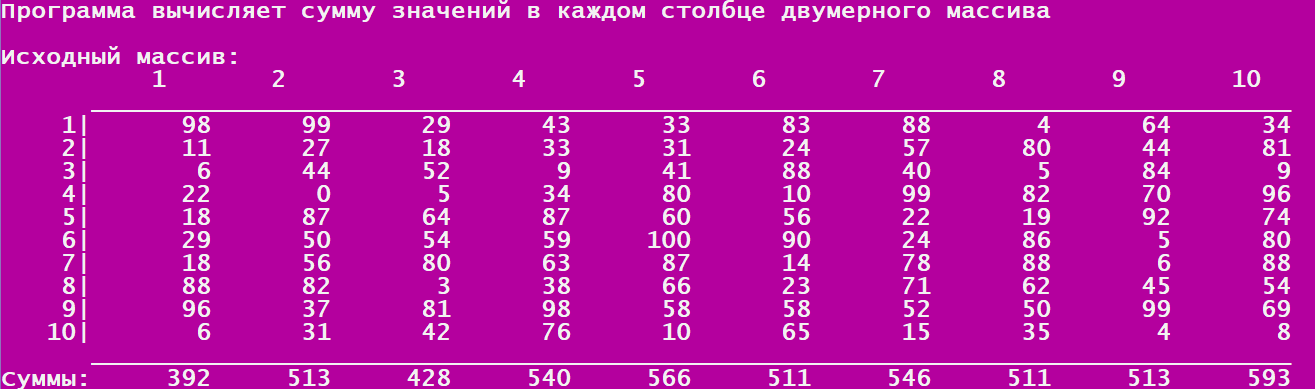
for (i = 0; i < 10; i++)

sum += arr[i][j];

printf("\t%6d", sum);

}

**Результат работы программы:**



**Вывод:**

Написали программу, вычисляющую сумму значений в каждом столбце двумерного массива со случайными числами.

**Задание 5:** написать программу, осуществляющую поиск среди структур student структуру с заданными параметрами (фамилией, именем и т.д.).

**Описание работы программы:**

Устанавливаем в поток вывода и вывода кодовую таблицу, поддерживающую русский язык.

Объявляем массив stud[10] типа struct student.

struct student

{

char famil[11];

char name[11], facult[6];

int Nomzach;

} stud[10];

Объявляем символьный массив param[22] для ввода названия параметра, по которому будет осуществляться поиск, и символьный массив search[11] для значения фамилии, имени или факультета. Объявляем переменную somebody типа int со значением 0. Она будет использоваться как флаг, на случай если студентов с данными, по которым выполняется поиск, не найдется. Если к концу программы её значение останется равным 0, значит, результатов поиска нет и требуется вывод соответствующего сообщения. Инициализируем структуру данными студентов.

Выводим на экран информацию о том, по каким параметрам можно выполнить поиск. Считываем в массив param введенное пользователем значение. Проверяем, не было ли введено невозможное значение. Пока пользователь не введет существующий параметр, на экран будет выводится сообщение об ошибки и перечень допустимых параметров.

Если введено «Номер зачетной книжки», считываем номер, который пользователь вводит следующим шагом, в переменную searchzach типа int, так как введенное значение должно быть целым числом. Иначе считываем введенную информацию в массив search, так как все остальные параметры - строки.

if (strcmp(param, zachcontrol)==0)

scanf\_s("%d", &searchzach);

else

gets\_s(search);

Далее осуществляем поиск. Проверяем, введен ли номер зачетной книжки, и если это так, то начинаем цикл поиска. Сравниваем поле Nomzach каждого элемента массива структур с введенным значением. Если находится соответствие, флаг получает значение 1, а все данные о студенте выводятся на экран.

if (searchzach != 0)

{

for (int i = 0; i < 10; i++)

if (stud[i].Nomzach == searchzach)

{

somebody = 1;

printf("%12s\t%12s\t%4s\t\t%d\n", stud[i].famil, stud[i].name, stud[i].facult, stud[i].Nomzach);

}

}

В случае если осуществляется поиск по фамилии, имени или факультету, проверяем соответственно поля структуры famil, name, facult. В случае соответствия так же изменяется значение флага и выводятся все данные о студенте.

if (strcmp(param, "Фамилия") == 0)

{

for (int i = 0; i < 10; i++)

if (strcmp(search, stud[i].famil) == 0)

{

somebody = 1;

printf("%12s\t%12s\t%4s\t\t%d\n", stud[i].famil, stud[i].name, stud[i].facult, stud[i].Nomzach);

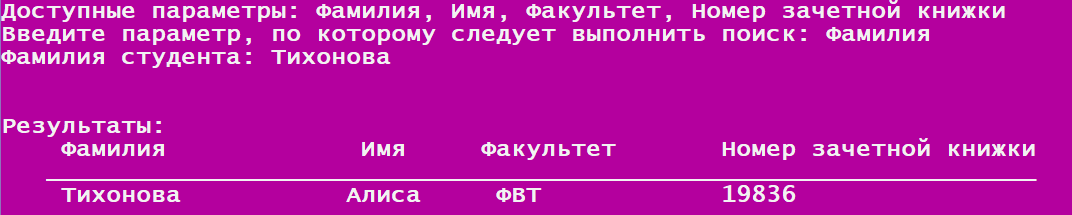
}

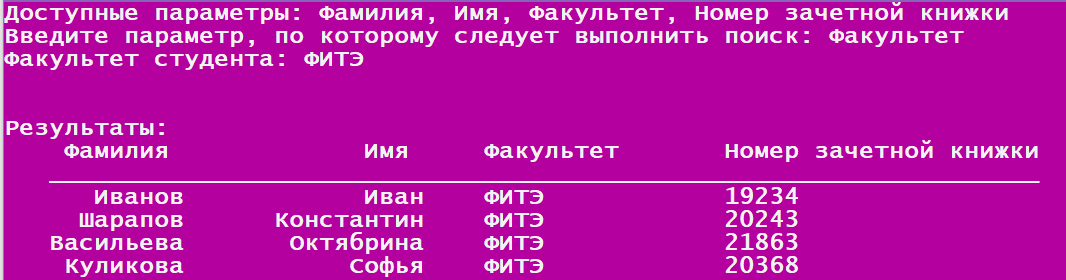
}

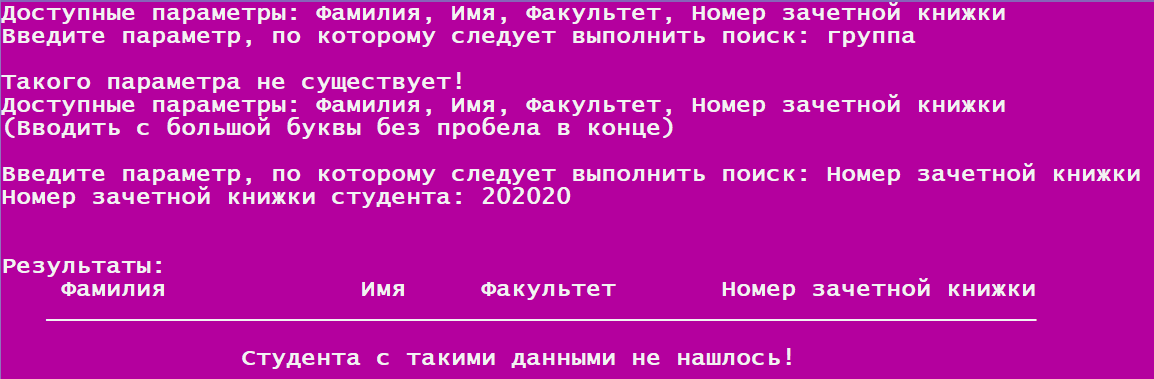
Если среди студентов есть те, у кого совпадает значение параметра, по которому выполняется поиск (например, студенты имеют одинаковую фамилию или учатся на одном факультете), выводятся все записи с таким значением параметра.

В конце проверяется значение флага. Если флаг равен нулю, выводится сообщение об отсутствии студентов с такими данными.

**Результаты работы программы:**

****





**Вывод:**

Написали программу, осуществляющую поиск среди структур student структуру с заданными параметрами (фамилией, именем и т.д.). В программе предусмотрели и обработали возможный ввод некорректных данных и ситуацию, при которой студентов с искомыми данными не существует.