Пензенский государственный университет

Кафедра ”Вычислительной техники ”

**Отчет**

по лабораторной работе №1

по дисциплине: “ Логика и основы алгоритмизации в инженерных задачах ”

## по теме: “Простые структуры данных”

## *Выполнили студенты группы 19ВВ1:*

## Чернов А.

## Манягин А.

## *Приняли:*

## Митрохин М.А

ПЕНЗА 2020

**Общие сведения.**

К простым структурам данных языка Си относятся массивы, строки (массивы символов). Структуры (struct) – простейшая составная структура данных.

**Массив** – это конечная совокупность данных одного типа.

Массивы могут состоять из целых чисел, чисел с плавающей запятой, символов и других типов данных. Существуют даже массивы массивов, – многомерные массивы.

Любой массив в программе должен быть описан: вначале указывается тип массива. затем его имя, после имени массива добавляют квадратные скобки ([ ]) внутри которых обычно стоит число, показывающее количество элементов массива.

**Динамический массив**

Для использования функций динамического распределения памяти необходимо подключение библиотеки <malloc.h> или <stdlib.h> (в зависимости от компилятора и используемого стандарта языка).

Функции динамического выделения памяти

void\* malloc(*РазмерМассиваВБайтах*);  
 void\* calloc(*ЧислоЭлементов*, *РазмерЭлементаВБайтах*);

выделяют блок памяти, размером *РазмерМассиваВБайтах* или *ЧислоЭлементов*\**РазмерЭлементаВБайтах* байт, и возвращает указатель на начало блока.

Элементами массива могут быть как данные простых типов, так и составных типов (например, структуры), поэтому для точного определения размера элемента массива в общем случае используется функция

int sizeof(*ТипЭлемента*);

Эта функция возвращает количество байт, занимаемое элементом указанного типа.

Память, динамически выделенная с использованием функций calloc(), malloc(), должна быть освобождена после окончания использования функцией

 free(*указатель*);

Для примера, код

**int** \*a; // указательнамассив

**int**n;

...

n = 10;

// Выделениепамяти

a = (**int**\*)malloc(n \* **sizeof**(**int**));

...

a[1] = a[1] + a[3];

...

free(a); // Освобождениепамяти

создаёт массив размером 10 элементов и после некоторых преобразований высвобождает занимаемую им память.

Для **генерации случайных чисел** в языке Си используется функция rand(), которая содержится в библиотеке <stdlib.h>.

Функция возвращает случайное целое число в диапазоне от нуля до RAND\_MAX. Глобальная константа RAND\_MAX является максимальным значением, которое может возвращаться функцией rand, это значение определено в заголовочном файле stdlib.h.

Чтобы ограничить сверху случайные числа, можно воспользоваться операцией получения остатка от деления. Остаток от деления на число всегда меньше этого числа. Поэтому для ограничения сверху случайных чисел возьмём остаток от деления на граничное значение.

intk;

k = rand()%100;

В приведенном примере значение k не превысит 99. Т.е. для rand()%n возвращается любое число в диапазоне [0, n). Если необходимо получить числа в диапазоне [0, n], то необходимо изменить выражение на rand()%(n+1).

Для получения чисел из некоторого отрезка [min, max] необходимо найти остаток от деления на длину этого отрезка и выполнить сдвиг возвращаемого остатка на нижнюю границу отрезка

intk;

k = min + rand()%(max-min+1);

При повторных запусках программы функция rand() будет возвращать одинаковые последовательности чисел. Для получения уникальных последовательностей необходимо инициализировать генератор случайных чисел различными значениями. Для этого используется функция srand(), аргументом которой является инициализирующее целое число.

В качестве уникального для каждого запуска программы числа чаще всего используется текущее время, возвращаемое функцией time(), прототип которой описан в библиотеке time.h. Передав в time() в качестве параметра NULL, мы получим целое число, которое можно передать в srand(), установив, таким образом уникальное инициализирующее значение генератора

srand(time(NULL));

int k;

k = rand();

теперь генерируемая последовательность чисел k будет уникальной.

**Структура** - это совокупность переменных одного или нескольких типов, сгруппированных в один элемент.

Объявление структуры осуществляется с помощью ключевого слова struct, за которым идет ее тип и далее список элементов, заключенных в фигурные скобки:

struct тип { тип элемента\_1 имя элемента\_1;

.........

тип элемента\_n имя элемента\_n;

};

Именем элемента может быть любой идентификатор.

Определениеструктуры:

struct student

{

intNomzach;

charfamil[20];

char name[20], facult[20];

};

После фигурной скобки, заканчивающей список элементов, могут инициализироваться структуры заданного типа, например:

structstudent{...} Ivanov, Petrov;

Доступ к отдельным полям структуры осуществляется с помощью оператора . (оператор точка). Например, доступ к полю name в структуре Ivanov осуществляется так:

Ivanov.name;

Код:

/\* Задание 1-3

1) выводит разницу между максимальным и минимальным числом

2) Устанавливаем размер массива в консоле

3) В массив выводит рандомные числа

#include <iostream>

#include <stdio.h>

#include <locale>

#include <malloc.h>

#include <stdlib.h>

#include <time.h>

#include <locale.h>

#include <windows.h>

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

srand(time(NULL));

int arr;

int max = 0;

int min = 0;

int sum = 0;

int \*b;

// Вводим размер массива

printf("Введите размер массива ");

scanf("%d", &arr);

// Выделяем память под массив

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

// Интелезируем массив

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

printf("%d\n", b[i] = rand() % 100);

// Вычисляем разницу

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

if (b[j] > max) {

max = b[j];

min = b[j] + 1;

}

}

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

if (b[j] < min) {

min = b[j];

}

}

sum = max - min;

// Выводим ответ

printf("Max = %d\n", max);

printf("Min = %d\n", min);

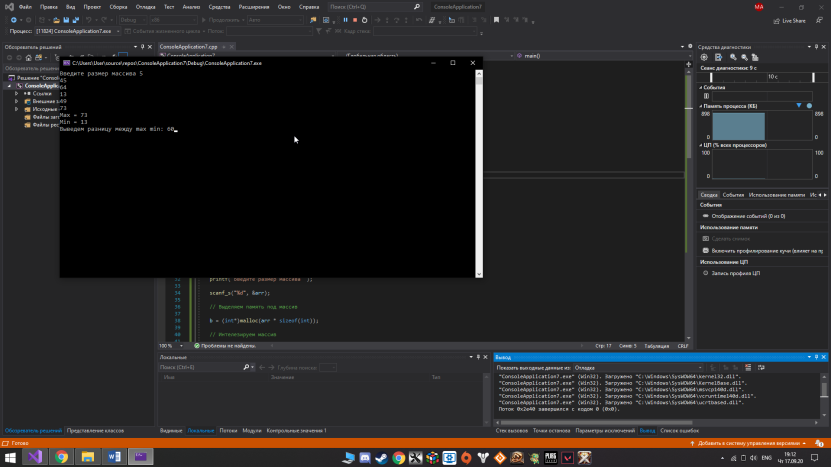
printf("Выведем разницу между max min: %d", sum);

free(b);

getchar(), getchar();

return 0;

}



Задание 4

#include <iostream>

#include <stdio.h>

#include <locale>

#include <malloc.h>

#include <stdlib.h>

#include <time.h>

#include <locale.h>

#include <windows.h>

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

int a[4][4];

int result = 0;

for (int i = 0; i< 4; i++) {

for (int j = 0; j < 4; j++) {

a[i][j] = rand() % 30;

printf("%d ", a[i][j]);

result += a[i][j];

}

printf (" = %d ", result);

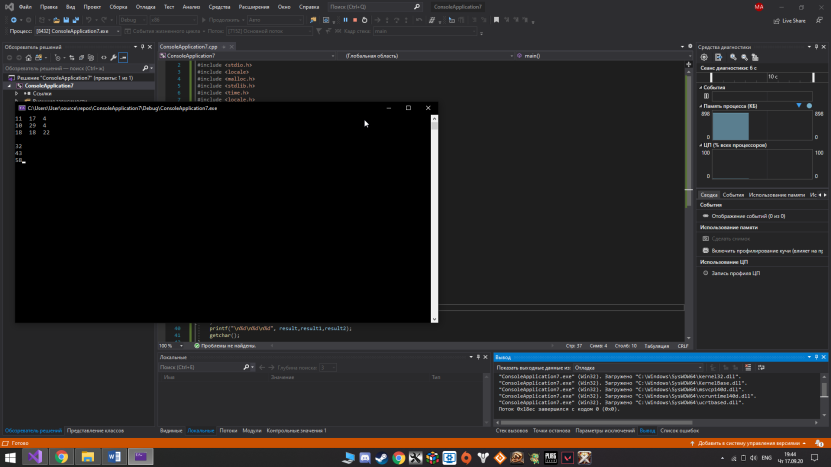
printf("\n");

result = 0;

}

getchar();

}



Задание 5

#include "stdafx.h"

#include <string.h>

#include <windows.h>

#include<locale.h>

int main(void)

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

setvbuf(stdin, NULL, \_IONBF, 0);

setvbuf(stdout, NULL, \_IONBF, 0);

int i;

struct student

{

char famil[20];

char name[20], facult[20];

int Nomzach;

} stud[2];

for(i=0;i<2;i++)

{

printf("Введите фамилию студента\n"); scanf ("%20s",stud[i].famil);

}

for(i=0;i<2;i++)

{

printf("Введите имя студента %s\n",stud[i].famil); scanf ("%20s",stud[i].name);

}

for(i=0;i<2;i++)

{

printf("Введите название факультета студента %s %s\n",stud[i].famil,stud[i].name); scanf ("%20s",stud[i].facult);

}

for(i=0;i<2;i++)

{

printf("Введите номер зачётной книжки студента %s %s\n",stud[i].famil,stud[i].name); scanf ("%d",&stud[i].Nomzach);

}

for(i=0;i<2;i++)

{

printf("Cтудент %s %s обучается на факультете %s, номер зачётной книжки %d \n",stud[i].famil,stud[i].name,

stud[i].facult,stud[i].Nomzach);

}

char s[80];

int g;

printf("Введите фамилию: ");

scanf("%s",s);

for( i=0; i<2; i++){

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

printf("Найденный Cтудент %s %s обучается на факультете %s, номер зачётной книжки %d \n",stud[i].famil,stud[i].name,

stud[i].facult,stud[i].Nomzach);

g=1;}

if (g ==0)

printf("Студент не найден");

getchar(); getchar();

}

