МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Казанский национальный исследовательский технический университет им.

А.Н. Туполева – КАИ»

Институт компьютерных технологий и защиты информации Отделение СПО ИКТЗИ (Колледж информационных технологий)

ЛАБОРАТОРНАЯ РАБОТА № 5

по дисциплине СИСТЕМНОЕ ПРОГРАММИРОВАНИЕ

Работу выполнил Студент гр.4335 Желваков А.С

Принял Преподаватель Григорьева В.В.

Казань 2024

**Цель работы** Приобрести умения и практические навыки для работы с указателями на функцию и передачей функций как параметров в другие функции; приобрести умения и практические навыки для работы со структурами.

# Задание 1

**1, 7 вариант:**

Задание 1. Написать функцию, которая возвращает значение

выражения.

1. f(x)=2\*x;

7. f(x)=(5-x)/x;

В таблице 1 представлен код решения задания

| #include <stdio.h>  // Функция для пункта 1: f(x) = 2x  double func1(double x) {  return 2 \* x;  }  // Функция для пункта 7: f(x) = (5 - x)x  double func7(double x) {  return (5 - x) \* x;  }  // Функция для вычисления значения через указатель на функцию  double calculate(double x, double (\*f)(double)) {  return f(x);  }  int main() {  double x;  printf("Введите значение x: ");  scanf("%lf", &x);  // Указатели на функции  double (\*pf1)(double) = func1;  double (\*pf7)(double) = func7;  // Вычисление и вывод результата для функции 1  double result1 = calculate(x, pf1);  printf("f(x) = 2x при x = %.2f: %.2f\n", x, result1);  // Вычисление и вывод результата для функции 7  double result7 = calculate(x, pf7);  printf("f(x) = (5 - x)x при x = %.2f: %.2f\n", x, result7);  return 0;  } |
| --- |

Таблица 1

На рисунке 1 показан результат выполнения программы

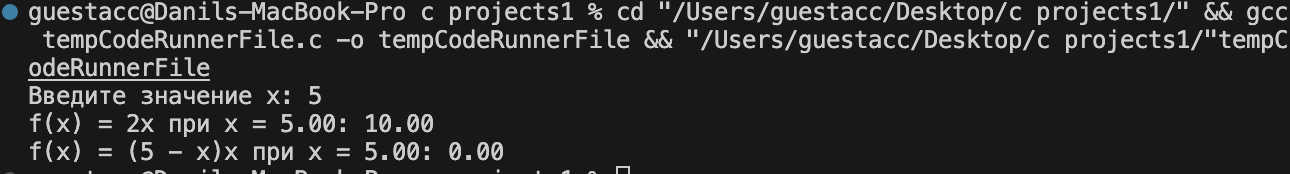


Рисунок 1

**Задание 2**

**6 вариант:**

Задание 2. Написать функцию Function, которая возвращает в вызывающую программу значение выражения в соответствии с вариантом. Здесь f1(a) и f2(a) – функции с одним параметром целого типа, которые возвращают целочисленное значение (используйте функции из задания 1). Передавать эти функции в Function при помощи указателя на функцию.Функция main выводит результаты функции Function.

6. 1/( f1(a)+f2(a) );. В таблице 3 представлен код решения задания.

| #include <stdio.h>  // Функция f1(a) = 2a (из предоставленного кода)  double f1(double a) {  return 2 \* a;  }  // Функция f2(a) = (5 - a)a (из предоставленного кода)  double f2(double a) {  return (5 - a) \* a;  }  // Функция для вычисления 1/(f1(a) + f2(a))  double Function(double a, double (\*pf1)(double), double (\*pf2)(double)) {  double result\_f1 = pf1(a);  double result\_f2 = pf2(a);  printf("%.2f result 1\n", result\_f1);  printf("%.2f result 2\n", result\_f2);    double sum = result\_f1 + result\_f2;  if (sum == 0) {  printf("Ошибка: деление на ноль при a = %.2f\n", a);  return 0;  }  return 1 / sum;  }  int main() {  double a;  printf("Введите число a: ");  scanf("%lf", &a);  double (\*pf1)(double) = f1;  double (\*pf2)(double) = f2;  double result = Function(a, pf1, pf2);  printf("1/(f1(a) + f2(a)) при a = %.2f: %.2f\n", a, result);  return 0;  } |
| --- |

Таблица 3

На рисунке 3 показан результат выполнения программы

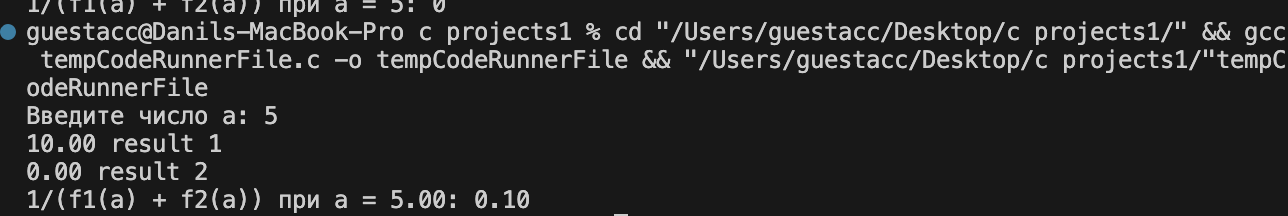


Рисунок 3

**Задание 3**

Задание 3. Написать программу, которая реализует меню из 4 функций, которые выполняют следующие действия: возведение числа a в степень b, произведение a и b, получение остатка от деления a на b, проверка кратности чисел a и b. Примечание: a и b – целые числа.

В таблице 4 представлен код решения задания

| #include <stdio.h>  int power(int a, int b) {  int result = 1;  for (int i = 0; i < b; i++) {  result \*= a;  }  return result;  }  int multiply(int a, int b) {  return a \* b;  }  int remainder(int a, int b) {  if (b == 0) {  printf("Ошибка: деление на ноль!\n");  return 0;  }  return a % b;  }  int is\_multiple(int a, int b) {  if (b == 0) {  printf("Ошибка: деление на ноль!\n");  return 0;  }  return (a % b == 0) ? 1 : 0; // 1 - кратно, 0 - не кратно  }  int calculate(int a, int b, int (\*func)(int, int)) {  return func(a, b);  }  int main() {  int a, b, choice;  int (\*operations[4])(int, int) = {power, multiply, remainder, is\_multiple};  do {  printf("\nМеню:\n");  printf("1. Возведение a в степень b\n");  printf("2. Произведение a и b\n");  printf("3. Остаток от деления a на b\n");  printf("4. Проверка кратности a и b\n");  printf("0. Выход\n");  printf("Выберите действие (0-4): ");  scanf("%d", &choice);  if (choice >= 1 && choice <= 4) {  printf("Введите a: ");  scanf("%d", &a);  printf("Введите b: ");  scanf("%d", &b);  int result = calculate(a, b, operations[choice - 1]);  switch (choice) {  case 1:  printf("%d в степени %d = %d\n", a, b, result);  break;  case 2:  printf("%d \* %d = %d\n", a, b, result);  break;  case 3:  if (b != 0) {  printf("%d %% %d = %d\n", a, b, result);  }  break;  case 4:  if (b != 0) {  printf("%d %s кратно %d\n", a, result ? "является" : "не является", b);  }  break;  }  } else if (choice != 0) {  printf("Неверный выбор! Введите число от 0 до 4.\n");  }  } while (choice != 0);  printf("Программа завершена.\n");  return 0;  } |
| --- |

На рисунке 4 показан результат выполнения программы

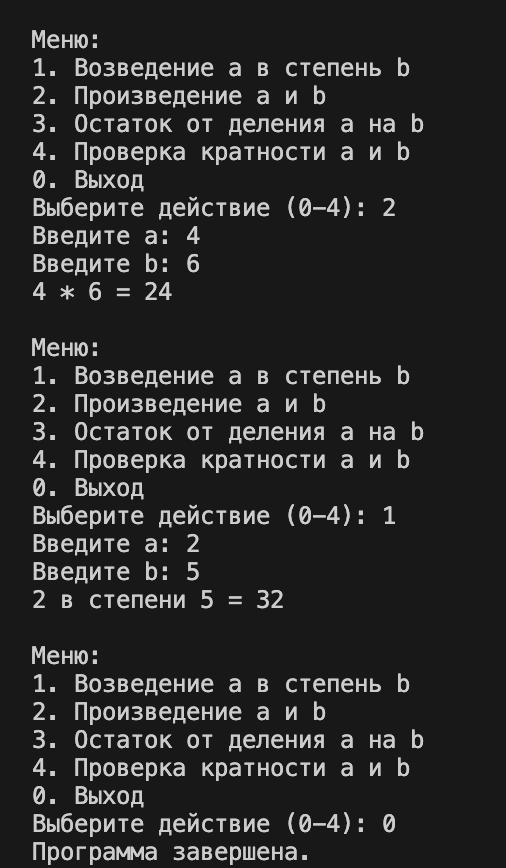


Рисунок 4

**Задание 4**

Задание 4. Создать и проинициализировать массив структур из 10 элементов, состоящих из следующих полей:

6. Поля структуры: название детали, цена детали, количество. Вывести на экран название самой дорогой детали. Массив структур проинициализирован при объявлении.

В таблице 4 представлен код решения задания

| #include <stdio.h>  // Структура для детали  struct Part {  char name[50]; // Название детали  double price; // Цена детали  int quantity; // Количество  };  // Функция для вывода самой дорогой детали  void print\_most\_expensive\_part(struct Part \*parts, int size) {  if (size <= 0) {  printf("Массив пуст\n");  return;  }  // Находим индекс самой дорогой детали  int max\_price\_index = 0;  for (int i = 1; i < size; i++) {  if (parts[i].price > parts[max\_price\_index].price) {  max\_price\_index = i;  }  }  // Выводим информацию о самой дорогой детали  printf("Самая дорогая деталь:\n");  printf("Название: %s, Цена: %.2f, Количество: %d\n",  parts[max\_price\_index].name,  parts[max\_price\_index].price,  parts[max\_price\_index].quantity);  }  int main() {  // Инициализация массива из 10 деталей  struct Part warehouse[10] = {  {"Шестерня", 150.50, 100},  {"Подшипник", 75.20, 200},  {"Вал", 300.00, 50},  {"Ремень", 45.90, 150},  {"Гайка", 5.30, 500},  {"Болт", 3.80, 600},  {"Пружина", 20.00, 300},  {"Клапан", 120.75, 80},  {"Фильтр", 60.40, 120},  {"Поршень", 200.25, 70}  };  // Указатель на функцию для вывода самой дорогой детали  void (\*print\_func)(struct Part \*, int) = print\_most\_expensive\_part;    // Вызов функции через указатель  print\_func(warehouse, 10);  return 0;  } |  |
| --- | --- |

На рисунке 5 показан результат выполнения программы

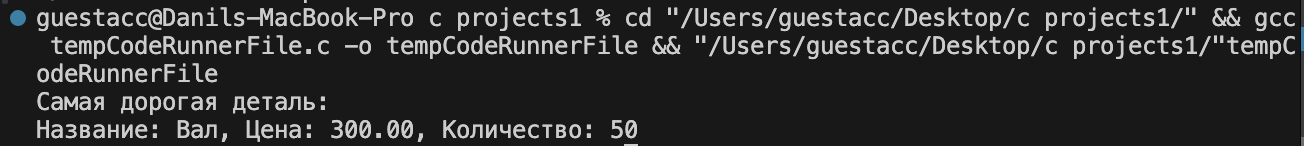


Рисунок 5

**Задание 5**

Задание 5. Создать и проинициализировать динамический массив структур из 5 элементов, состоящих из следующих полей:

3. Поля структуры: имя, фамилия, отчество, возраст. Написать функцию, которая заполняет поля структуры значениями, вводимыми с клавиатуры, и возвращает структуру в вызывающую программу. Вывести заполненную структуру на экран.

В таблице 4 представлен код решения задания

| #include <stdio.h>  #include <stdlib.h>  // Структура для человека  struct Person {  char name[50]; // Имя  char surname[50]; // Фамилия  char patronymic[50]; // Отчество  int age; // Возраст  };  // Функция для заполнения структуры  struct Person fill\_person() {  struct Person person;  printf("Введите имя: ");  fgets(person.name, 50, stdin);  for (int i = 0; i < 50; i++) {  if (person.name[i] == '\n') {  person.name[i] = '\0';  break;  }  }  printf("Введите фамилию: ");  fgets(person.surname, 50, stdin);  for (int i = 0; i < 50; i++) {  if (person.surname[i] == '\n') {  person.surname[i] = '\0';  break;  }  }  printf("Введите отчество: ");  fgets(person.patronymic, 50, stdin);  for (int i = 0; i < 50; i++) {  if (person.patronymic[i] == '\n') {  person.patronymic[i] = '\0';  break;  }  }  printf("Введите возраст: ");  scanf("%d", &person.age);  getchar(); // Очистка буфера после scanf  return person;  }  // Функция для вывода массива структур  void print\_people(struct Person \*people, int size) {  printf("\nСписок людей:\n");  for (int i = 0; i < size; i++) {  printf("Человек %d:\n", i + 1);  printf(" Имя: %s\n", people[i].name);  printf(" Фамилия: %s\n", people[i].surname);  printf(" Отчество: %s\n", people[i].patronymic);  printf(" Возраст: %d\n", people[i].age);  }  }  int main() {  int size = 5;  struct Person \*people = (struct Person \*)malloc(size \* sizeof(struct Person));  if (people == NULL) {  printf("Ошибка выделения памяти!\n");  return 1;  }  for (int i = 0; i < size; i++) {  printf("\nЗаполнение данных для человека %d:\n", i + 1);  people[i] = fill\_person();  }  // Указатель на функцию для вывода  void (\*print\_func)(struct Person \*, int) = print\_people;  print\_func(people, size);  free(people);  return 0;  } |  |
| --- | --- |

На рисунке 5 показан результат выполнения программы

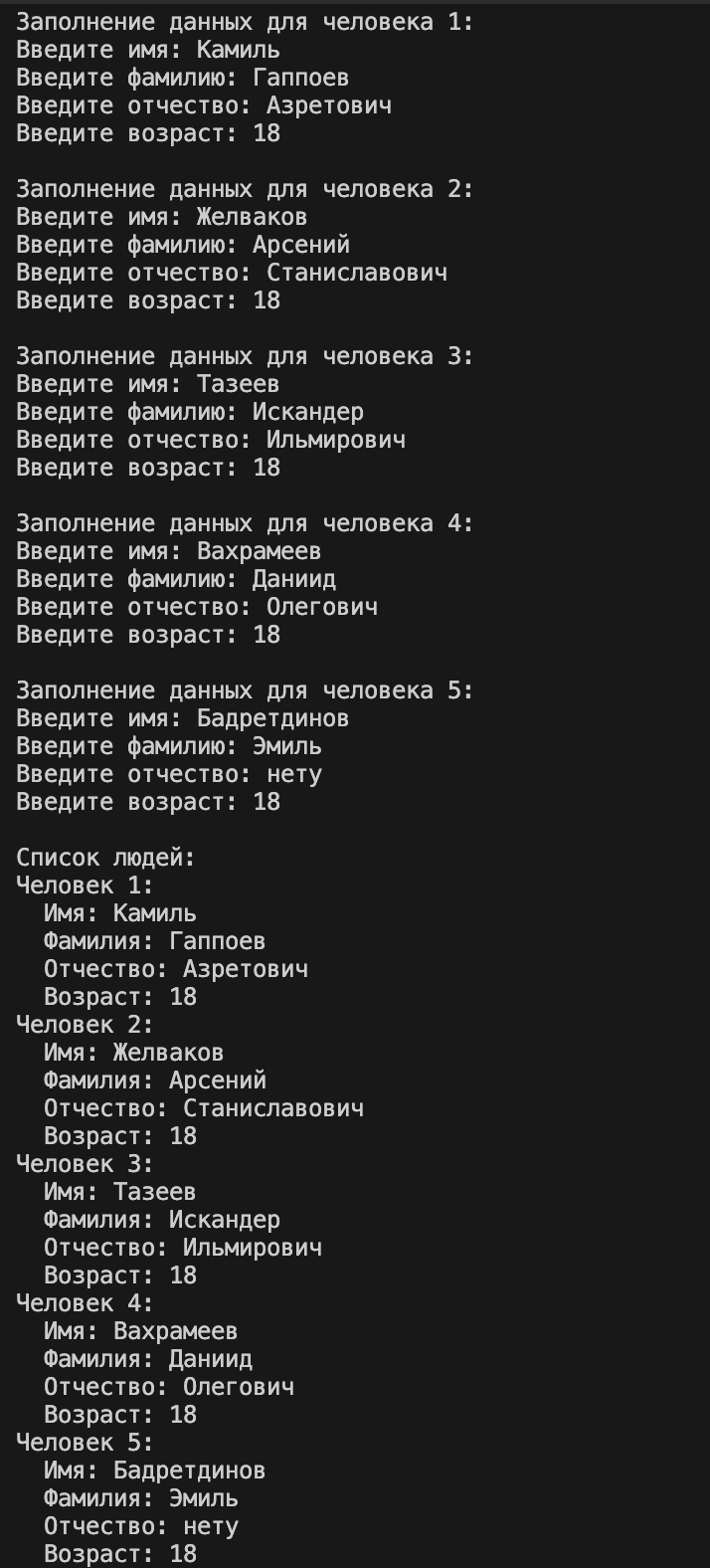


Рисунок 5

**Вывод:** Я приобрёл умения и практические навыки для работы по созданию динамических массивов, по работе с функциями и указателями.