Отчёт по лабораторной работе №6

Рекурсивные функции

Задание:

1). Для решения задачи составить: не рекурсивную функцию, рекурсивную функцию, возвращающую значение, рекурсивную функцию типа **void**. Объединить функции в одной программе и сравнить результаты.

|  |
| --- |
| 15. Дано натуральное число N. Вычислите сумму его цифр. |

2). В задании№1 Лабораторной работы №1 (этот семестр) функции ввода и вывода массива сделать рекурсивными без использования циклов.

|  |
| --- |
| 15. Описать структуру AEROFLOT, содержащую поля: название пункта назначения, дата вылета, время вылета. Реализовать ввод данных с клавиатуры в массив, состоящий из 5 структур типа AEROFLOT. Вывести на экран даты вылета и время вылета самолетов, вылетающих в пункт назначения, название которого совпало с названием, введенным с клавиатуры. Если таких рейсов нет, то вывести соответствующее сообщение. |

Код программы:

|  |
| --- |
| #include <iostream>  #include <sstream>  int \_sum(int number) {  int sum = 0;  while (number % 10 != 0) {  sum += number % 10;  number /= 10;  }  return sum;  }  int rec\_sum(int number) {  int sum = 0;  if (number != 0) {  sum = rec\_sum(number / 10);  sum += number % 10;  }  return sum;  }  void rec\_sum(int number, int& sum) {  sum = 0;  if (number != 0) {  rec\_sum(number / 10, sum);  sum += number % 10;  }  }  struct AEROFLOT  {  std::string destination;  std::string out\_date;  std::string out\_time;  };  void structgen(int pos, int number, AEROFLOT\*\* arr) {  std::string buff;  std::getline(std::cin, buff);  if (pos < number) {  int i = pos + 1;  arr[pos] = new AEROFLOT;  std::cout << "[Самолёт " << i << "] Введите место назначения: "; std::getline(std::cin, arr[pos]->destination);  std::cout << "[Самолёт " << i << "] Введите дату и время вылета самолёта: "; std::cin >> arr[pos]->out\_date >> arr[pos]->out\_time;  structgen(pos + 1, number, arr);  }  }  void structout(int pos, int number, AEROFLOT\*\* arr, std::string needed, int &sum) {  if (pos < number) {  if (arr[pos]->destination == needed) {  int i = pos + 1;  std::cout << "Самолёт " << i << " вылетает в " << arr[pos]->out\_date << " " << arr[pos]->out\_time << std::endl;  sum++;  }  structout(pos + 1, number, arr, needed, sum);  }  if (!sum) {  std::cout << "Самолёты не найдены!" << std::endl;  sum = -1;  }  }  int main() {  setlocale(LC\_ALL, "Russian");  int prog;  std::cout << "Выберите задание [1/2]: "; std::cin >> prog;  switch (prog) {  case 1: {  int num;  std::cout << "int = ";  std::cin >> num;  int sum = 0;  sum = \_sum(num);  std::cout << "[Обычная] sum = " << sum << std::endl;  sum = rec\_sum(num);  std::cout << "[Рекурсивная] sum = " << sum << std::endl;  rec\_sum(num, sum);  std::cout << "[Рекурсивная void] sum = " << sum << std::endl;  break;  }  case 2: {  const int n = 5;  int sum = 0;  AEROFLOT\* arr[n];  structgen(0, n, arr);  std::string needed;  std::cout << "Пункт назначения: "; std::getline(std::cin, needed);  structout(0, n, arr, needed, sum);  break;  }  }  return 0;  } |

Алгоритм работы:

Задание 1:

Программа запрашивает у пользователя целое число, после чего вызывает 3 функции с единственным параметром – запрошенное число. Все 3 функции возвращают сумму цифр числа, но отличается реализация этих функций:

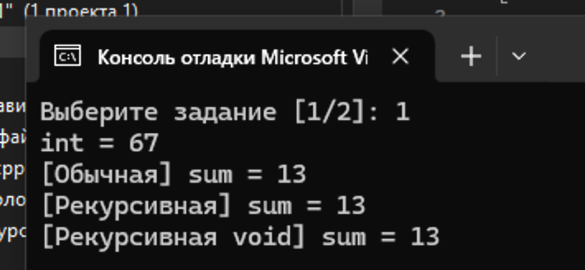
* Первая реализована как функция, возвращающая значение – вводится переменная, к которой при помощи цикла прибавляется каждая цифра числа. Конечное значение суммы возвращается в то место, где функция была вызвана.
* Вторая функция реализована как рекурсивная функция, возвращающая значение. Создаётся переменная для хранения суммы, затем функция вызывает сама себя, передав в качестве параметра исходное число без последней цифры. После этого от переданного числа отделяется цифра и суммируется к созданной в каждой функции сумме. Начиная с последней, функции возвращают сумму в предыдущую функцию, а затем изначально вызванная функция возвращает значение туда, откуда её вызвали.
* Третья функция реализована как рекурсивная функция, но в отличие от предыдущей, функция не возвращает значение, а работает с ссылкой на переменную с суммой. Отличием от предыдущей будет дополнительный параметр ссылка при вызове функции.

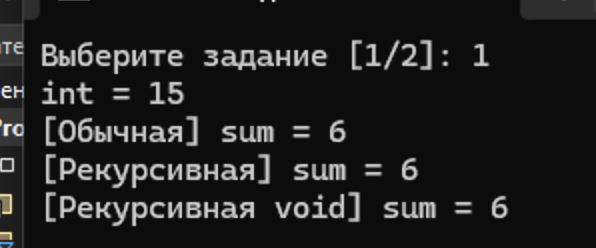
Задание 2:

Программа создаёт массив структур из 5 элементов. Затем вызывается рекурсивная функция, которая запрашивает у пользователя данные о рейсах: место назначения, дата/время. Функция принимает 3 параметра: порядковый номер самолёта, размер массива структур и сам массив структур. При вызове функция проверяет меньше ли порядковый номер размера массива, если меньше, то функция запрашивает данные у пользователя, записывает их в структуру и вызывает саму себя с увеличенным на 1 параметром порядкового номера. После программа запрашивает искомое место назначения и вызывает ещё одну рекурсивную функцию с параметрами: порядковый номер, размер массива, массив структур, искомое место, сумма рейсов. Та, в свою очередь, проверяет меньше ли порядковый номер размера и совпадает ли для рейса место назначения с искомым. Если условия выполнены, функция печатает рейс на экране и увеличивает сумму подходящих рейсов, а затем вызывает сама себя. Если после окончания вызовов функцией самой себя, сумма рейсов осталась нулевой, то функция присваивает сумме значение -1 и печатает на экране сообщение об ошибке.

Результаты работы:

Задание 1:





Задание 2:

