Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Лабораторная работа 10

По дисциплине «Основы алгоритмизации и программирования»

На тему «Рекурсивные алгоритмы»

Выполнила:

Студентка 1 курса 2 группы

Глухова Д.В.

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 2

4. В соответствии со своим вариантом выполнить задания из таблицы, представленной ниже. В некоторых заданиях имеются ошибки: не выполняется условие завершения рекурсии. Изменить условие такой задачи с тем, чтобы рекурсия выполнялась.

5. К номеру своего варианта прибавить число 2 и написать программу для новых исходных данных (для вариантов 15, 16 перейти к вариантам 1, 2).

Дополнительные задания

1. Ввести цифру **А**, записать в файл все возможные числа, состоящие из цифр, не превышающих или равных **A**. Количество цифр в числах должно быть равно **А**.

Примечание: использовать дополнительный массив.

3. Дано **n** различных натуральных чисел (**n = 5**). Напечатать все перестановки этих чисел.

4. По заданному числу **n** определить символ, который стоит на **n**-ом месте последовательности, получившейся после шага c номером 26.

|  |
| --- |
| Исходный код |
| #include <iostream>  #include <fstream>  using namespace std;  // функция для определения максимального элемента  unsigned int maxFunc(unsigned int n, unsigned int m){  int max;  if (m > n)  max = m;  else  max = n;  return max;  }  // функция второго варианта  unsigned int taskFunction(unsigned int n, unsigned int m) {  if ((m + n) % 2 == 0)  {  return maxFunc(n, m);  }  else {  return (taskFunction((n + m + 1) / 2, n + 1) + taskFunction(m, (m + n + 1) / 2) );  }  }  // дополнительная функция для четвертого варианта  double mod(int a, int b) {  return (a - floor(a / b) \* b);  }  // функция четвертого варианта  double taskTwo(double a, double b){  if (b == 0) return 0;  if (b != 0) {  return ( floor(a / b) + taskTwo(b, mod(a, b) ) );  }  }  // 1. Ввести цифру А, записать в файл все возможные числа, состоящие из цифр, не превышающих или равных A. Количество цифр в числах должно быть равно А.  // Примечание: использовать дополнительный массив.  void dop1(int size, int\* arr, int tmp)  {  ofstream file("data.txt", ios\_base::app);  for (int i = 0; i <= size; i++) {  arr[tmp] = i;  if (tmp == size - 1) {  for (int j = 0; j < size; j++)  {  cout << arr[j];  file << arr[j];  }  cout << " ";  file << " ";  }  else  dop1(size, arr, tmp + 1);  }  }  char c[50] = {};  char tr[50] = {};  int count = 0;  // 3. Дано n различных натуральных чисел (n = 5). Напечатать все перестановки этих чисел.  void dop2(string s, int count)  {  if (count == strlen(c))  {  cout << s << endl;  return;  }  for (int i = 0; i < strlen(c); i++) // цикл для перестановки чисел в разном порядке  {  if (!tr[i])  {  count++;  string save = s;  s += c[i];  tr[i] = 1;  dop2(s, count);  count--;  s = save;  tr[i] = 0;  }  }  }  // 4. По заданному числу n определить символ, который стоит на n - ом месте последовательности, получившейся после шага c номером 26.  static char dop3(char\* arr, const int& n, int m) {  if (m < 26) dop3(arr, n, m + 1);  if (m == 26) return arr[n + m];  }  int main() {  setlocale(LC\_ALL, "rus");  int choose;  double a, b;  unsigned int m1, n1;  cout << "Выберите задание: \n";  cout << "1 - основа(2 вар)\n";  cout << "2 - основа2(4 вар)\n";  cout << "3 - доп1\n";  cout << "4 - доп2\n";  cout << "5 - доп3\n";  cout << "0 - выход\n";  do {  cout << "Выберите задание: \n";  cin >> choose;  switch (choose)  {  case 1: {  cout << "m = ";  cin >> m1;  cout << "n = ";  cin >> n1;  cout << "Результат = " << taskFunction(m1, n1) << "\n";  break;  }  case 2:  cout << "a = ";  cin >> a;  cout << "b = ";  cin >> b;  cout << "Результат = " << taskTwo(a, b) << " квадратиков" << "\n";  break;  case 3: {  int A;  cout << "A = ";  cin >> A;  int\* arr = new int[A];  if (A == 1)  {  cout << 0 << " ";  cout << 1 << '\n';  }  else  for (int i = 1; i <= A; i++)  {  arr[0] = i;  dop1(A, arr, 1);  }  break;  }  case 4: {  for (int i = 0; i < 5; i++)  {  cout << "\nЧисло:\n";  cin >> c;  cout << "Перестановки:\n";  dop2("", 0);  }  break;  }  case 5: {  int max = 26;  constexpr int size = 52;  char array[size];  int n = 0;  for (int i = 0; i < max; i++)  {  array[i] = 'a' + i; // прибавляем к символу а число, чтобы получить след символ  array[i + max] = 'a' + i;  }  do  {  cout << "n = (n < 26 && n > 0)";  cin >> n;  for (int i = 0; i < size; i++) {  cout << i + 1 << ". " << array[i + 1] << endl; // выводим массив символов  }  cout << endl;  cout << "символ = " << dop3(array, n, 0) << endl;  } while (n);  break;  }  default:  break;  }  } while (choose != 0);  } |
| Консоль отладки |
| Задание 1 |
|  |
| Задание 2 |
|  |
| Дополнительное задание 1 |
|  |
| Дополнительное задание 2 |
| Дополнительное задание 3 |
|  |