|  |
| --- |
| Студент: Пантюхин А.Е., группа: ДТ-460а |
| Лабораторная работа №1 |
| Задание:  Дано n различных символов. Напечатать все возможные перестановки этих символов. |
| Тестовые данные:   1. 123 2. 323 3. 4Eaf 4. 0 |
| Словесное описание этапов решения задачи рекурсивным способом:   1. Функция принимает на вход два указателя, при первом вызове – одинаковых 2. Если входные данные не ведут на строки – выход 3. Первый указатель всегда обозначает начало строки, второй – текущий зафиксированный элемент 4. Если зафиксированный элемент – конец строки, то вывести строку 5. Иначе, начать движение по строке от зафиксированного указателя вправо 6. В цикле (пока не дошли до конца строки) 7. Меняем местами текущий элемент с текущим зафиксированным 8. Рекурсивный вызов функции, зафиксированный элемент – следующий от текущего зафиксированного 9. Обратно меняем местами значения зафиксированного и текущего элемента   В чём смысл:   1. Поочерёдно фиксируется каждый из элементов строки (рекурсивные вызовы) 2. Поочерёдно меняется каждый из элементов с зафиксированным(внутри цикла) 3. Таким образом, в рамках вызовов, сначала выведется неизменённая строка 4. Потом строка с перестановкой последних двух элементов 5. Далее третий с конца элемент поменяется местами со вторым с конца 6. Аналогично, строка с перестановкой последних двух 7. И так далее. |

|  |
| --- |
| Блок-схема алгоритма рекурсивной функции: |
| Текст программы с комментариями:  #include <stdio.h>  #include <locale.h>    //вспомогательная функция смены значений местами.  void swap\_values(char\* ptr1, char\* ptr2)  {  char buff = \*ptr1;  \*ptr1 = \*ptr2;  \*ptr2 = buff;  return;  }    //вывод перестановок  //source - указатель на начало строки, всегда  //locked\_ptr - указатель на "зафиксированный" элемент  int print\_permutations(char\* const source, char\* const locked\_ptr)  {  char\* move\_ptr = locked\_ptr; //указатель на адрес, начиная от текущего зафиксированного  //если на входе что-то не так - вернём ошибку  if (!source || !locked\_ptr) return 0;    //если служебный указатель дошёл до конца строки - выведем итоговую строку  if (!\*locked\_ptr) printf("%s\n",source);  //начиная с позиции служебного указателя(при первом вызове - 0, далее - по элементам строки)  //пока строка не кончилась  while (\*move\_ptr)  {  //поменяем местами значения по указателям (при первом вызове ничего не изменится)  swap\_values(locked\_ptr, move\_ptr);  //рекурсивный вызов со сдвигом постоянной позиции на 1 (позиции "зафиксированного" элемента)  print\_permutations(source, locked\_ptr+1);  //поменяем местами обратно для следующего вызова.  swap\_values(locked\_ptr, move\_ptr);  //движемся по строке далее  move\_ptr++;  }  return 1;  }    int main() {  setlocale(LC\_ALL, "Russian");  // BUFSIZ - директива препроцессора из библиотеки stdio, определяющая размер буфера потока.  char user\_input[BUFSIZ] = {0};  printf("Введите символы, не более %d:\n", BUFSIZ);  scanf("%s", user\_input);  //или fgets(user\_input, BUFSIZ-1, stdin);  print\_permutations(user\_input, user\_input);  return 0;  } |