Первое задание сделано (ниже код), но с небольшой ошибкой – выбрасывает программу сразу после запуска после нажатия любой клавиши. Просто нужно что-то добавить. Может getch(); не знаю...

1. «Текстовые файлы»

Имеется два текстовых файла. В первом из них содержится некоторое описание. Переносы слов допускаются. Второй файл содержит список слов, не подлежащих разглашению. Требуется переписать первый файл, заменив каждое из подобных слов точками.

# «Исследование состояний стека и очереди при выполнении над ними различных операций доступа»

В некотором институте приобретаемые компьютеры выделяются различным факультетам поочередно. В пределах факультета имеются очереди из кафедр. Факультет, получивший компьютер, перемещается в конец очереди, а соответствующая кафедра исключается из факультетской очереди. Вновь организованные факультеты и кафедры занимают последние места в соответствующих очередях. Составить программу ведения очереди на компьютеры.

1. «Исследование затрат времени на выполнение операций вставки и исключение элементов в древовидных структурах»

Ввести произвольное сильно ветвящееся дерево. Выдать списки вершин:

 1) являющихся листьями;

 2) не являющихся листьями;

 3) родителей листьев;

 4) заданного уровня считая от вершины.

1. «Методы сортировки»

Выбрать метод и написать программу сортировки телефонного справочника по фамилиям в лексикографическом порядке.

# «Алгоритмы быстрого поиска»

Составить программу удаления записи из сбалансированного бинарного дерева поиска (АВЛ-дерева).

1. «Алгоритмы на графах»

На плоскости заданы координаты N элементов, являющихся выводами печатной платы. Некоторые элементы связаны между собой. Требуется выбрать начальный элемент для построения контура (пути, не содержащего циклов) максимальной длины. Контур строится по следующему правилу: из всех возможных элементов, связанных с данным, в контур включается ближайший элемент.

**Код первого задания:**

#include "stdafx.h"

#include "string.h"

#include <conio.h>

#include <fstream>

#include <windows.h>

#pragma warning(disable: 4996)

const int SLOVOLEN = 128;

void soob(char msg[255]);

struct words {

 char slovo[SLOVOLEN];

 words \*next;

};

int IsAlphaNum(char c) // ANSI 1251

{

 if (c >= 'А' && c <= 'Я' || c >= 'а' && c <= 'я' || c >= 'A' && c <= 'Z' || c >= 'a' && c <= 'z' || c >= '0' && c <= '9') {

 return c;

 };

 return 0;

};

int fillsequence(words \*seq, char \*filename)

{

 FILE \*fs = fopen(filename, "r");

 char c;

 int i;

 memset(seq, 0, sizeof(words));

 while(!feof(fs)){

 i = fscanf (fs, "%c", &c);

 if (i > 0)

 if (IsAlphaNum(c) == 0){

 if (seq->slovo[0] != 0) {

 seq->next = new words;

 seq = seq->next;

 memset(seq, 0, sizeof(words));

 }

 }

 else {

 seq->slovo[strlen(seq->slovo)] = c;

 }

 }

 fclose(fs);

 return 0;

};

int freesequence(words \*dict)

{

 if (dict->next != NULL) {

 freesequence(dict->next);

 };

 delete dict;

 return 0;

};

int findwords(words \*dict, char \*str)

{

 if (dict != NULL) {

 if ( strcmp(dict->slovo, str) != 0) {

 return findwords(dict->next , str);

 }

 else {

 return 1;

 }

 };

 return 0;

}

int modifer(words \*dict, char \*filename\_in, char \*filename\_out)

{

 FILE \*fs = fopen(filename\_in, "r"); // file source

 FILE \*fd = fopen(filename\_out, "w"); // file destination

 char c;

 int i;

 char buf[SLOVOLEN];

 memset(buf, 0, SLOVOLEN);

 while(!feof(fs)){

 i = fscanf (fs, "%c", &c);

 if (IsAlphaNum(c) == 0 && i > 0){

 if (findwords(dict, buf) > 0) {

 fwrite("....", 1, 4, fd);

 }

 else {

 fwrite(buf, 1, strlen(buf), fd);

 };

 fwrite((char \*)(&c), 1, 1, fd);

 memset(buf, 0, SLOVOLEN);

 }

 else {

 buf[strlen(buf)] = c;

 }

 }

 fclose(fs);

 fclose(fd);

 return 0;

}

int main(int argc, char \*argv[])

{

 SetConsoleCP(1251);

 SetConsoleOutputCP(1251);

 if (argc < 2) soob("Не указан исходный файл");

 if (argc < 3) soob("Не указан файл словаря");

 if (argc < 4) soob("Не указан выходной файл");

 else if (argv[1] == NULL)

 soob("Невозможно открыть исходный файл");

 else if (argv[2] == NULL)

 soob("Невозможно открыть файл словаря");

 words \*dictionary = new words;

 fillsequence(dictionary, argv[2]);

 modifer(dictionary, argv[1], argv[3]);

 freesequence(dictionary);

 return 0;

}

void soob(char msg[255])

{

printf("%s\n",msg);

getch();

exit(0);

}