Министерство науки и образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Волгоградский государственный технический университет»

КАЧЕСТВО И НАДЕЖНОСТЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Программа для
«поиска минимального количества ходов, необходимых для получения искомой комбинации».

Внутренняя Спецификация

СОГЛАСОВАНО:
Руководитель проекта:
Доцент кафедры ПОАС
_____Сычев О. А.
«___»_____2021 г.

Разработчик:
Студент группы
ПрИн-267
_____Айоделе В. Д
«___»_____2021 г.

Нормоконтролер:
Преподаватель кафедры ПОАС
_____Матюшечкин Д.С.
«___»_____2021 г.

Волгоград, 2021 г.

Содержание

1. Общие сведения

Наименование программы: «Программа для поиска минимального количества ходов, необходимых для получения желаемой комбинации».

Для корректного функционирования программы необходима операционная система Windows 7 или выше. Дополнительного ПО не требуется.

Программа написана на языке C++.

2. Функциональное назначение

Программа предназначена для поиска минимального количества ходов, необходимых для получения желаемой комбинации.

3. Описание логической структуры

Программа считывает входные данные (подробный алгоритм в приложении Б). Программа работает путем замены значений из входной комбинации, чтобы получить желаемую комбинацию с минимальным количеством шагов (подробный алгоритм в приложении Б).

Результат записывается в указанный выходной файл (подробный алгоритм в Приложении Б).

4. Декомпозиция программы

- Основные типы и структуры данных программы представлены в приложении А.
- Спецификации функций программы представлены в приложении Б.
- Иерархия вызовов подпрограмм представлена в приложении В.
- Диаграмма потоков данных представлена в приложении Г.

5. Вызов и загрузка

Программа запускается посредством командной строки с указанием таких параметров как:

- имя файла с синтаксически верными выражениями на языке C++;
- имя текстовых файлов с ценами на бензин в каждом городе и маршрутами между городами.

Пример вызова программы из командной строки:

app.exe input.txt output.txt

6. Входные и выходные данные

Формат входных и выходных данных описан в пункте 3.3 и приложения А технического задания.

Приложение А

Основные типы и структуры данных программы

```
struct Combination {
	//массив char для хранения комбинации
	char data[2][4];

	//считываем из stdin комбинацию символов
	void read(istream& inputData) {
		for (int i = 0; i < 2; ++i) {
			for (int j = 0; j < 4; ++j) {
				inputData >> data[i][j];
			}
		}
	}

	//ищем '#', проверяем возможен ли сдвиг в di, dj и получаем новую комбинацию
	Combination shift(int di, int dj) const {
		Combination next = *this;
		for (int i = 0; i < 2; ++i) {
			for (int j = 0; j < 4; ++j) {
				if (next.data[i][j] == '#') {
					int ni = i + di;
					int nj = j + dj;
					if (0 <= ni && ni < 2 && 0 <= nj && nj < 4) {
						//если не выходим за края, то обменяем местами '#' и текущий символ и получим новую комбинацию
						swap(next.data[i][j], next.data[ni][nj]);
					}
					return next;
				}
			}
		}
	}
};
```

Приложение Б

Спецификации функций программы и Выделенные подпрограммы

```
int main(int argc, char* argv[]);
```

Входные данные:

- argc - количество аргументов командной строки, должно быть равно 3;
- argv - массив строк - аргументов командной строки, где
- argv[1] – путь к *.txt файла с оригинальной и необходимой комбинацией
- argv[2] – путь к *.txt файла где будет храниться результат.

Выходные данные:

- файл txt в котором содержащий сообщение о выполнении программы

Алгоритм:

1) если указаны не все параметры командной строки – записать ошибку и закончить работу программы;
2) получить абсолютный пути к файлам;
3) если невозможно создать или открыть указанный выходной файл;

   3.1. записать что неверно указан файл для выходных данных. Возможно указанного расположения не существует;
4) считать входные данные из файла
5) при удачном чтении файлов;

   5.1.1. если он не пуст, сохраняет извлеченные данные в массив и сравнивает их с требуемой комбинацией записать в выходной файл

   5.1.2. завершить работу программы;
6) при неудачном чтении файлов- завершить работу программы;

```
/*
\param[in] Combination& l – первая комбинация для сравнения
\param[in] Combination& r - вторая комбинация для сравнения

*/
```

int compare(const Combination& l, const Combination& r)

Алгоритм:

сравниватель двух комбинаций, если равны то возвращает 0, если l < r то возвращает отрицательны значение, если l > r возвращает положительное значение

```
/*! Считывает данные с входногофайла
 \return успешность считаниефайла
 */
```

//реализация решения
int solution(Combination& start, Combination& finish)

Алгоритм:

Любая ошибка прерывает выполнение алгоритма.

- Создать словарь для хранения соответствия комбинации->количеству шагов за которое в эту комбинацию пришли от начальной

- Создать очередь для хранения перебираемых комбинаций

- Пока очередь комбинаций не пуста

  - берем текущую комбинацию из очереди и проверяем, если она финальная то возвращаем количество шагов до нее и завершаем программу

- Генерируем новые комбинации сдвигая '#' во все возможные положения из текущего

- Очередь комбинаций опустела, но нужная не нашлась, выводим -1 по условию задач

Приложение В

Дерево вызовов функций