Внешняя спецификация

1. Назначение

Программа предназначена для проверки, объявлена ли переменная в коде программы на языке C.

2. Функциональные требования

2.1. Действия над объектами

Программа должна находить переменные в коде программы.

Программа должна уметь различать переменные от функций.

Программа должна определять объявлена ли переменная.

2.2. Ограничения

1)Возможные типы данных:

1.1) целочисленные типы данных: int, unsigned int, short, unsigned short, long, unsigned long, long long, unsigned long long;

1.2) типы данных с плавающей точкой: float, double, long double;

1.3) символьные типы данных: char;

1.4) перечисления enum;

1.5) структуры struct;

1.6) объединения union.

2)Код вводимой программы является компилируемым;

3)Поддерживаются глобальные и локальные переменные;

4)Не поддерживается директива #define;

5)Максимальное кол-во строк в программе - 100;

6)Максимальная длина строки - 100 символов;

7)Максимальное кол-во переменных - 100;

8)Поддерживается оператор const;

9)В коде могут встречаться однострочные и многострочные комментарии;

10)Поддерживаются указатели и массивы;

11)Не поддерживается typedef;

12)Поддерживаются анонимные структуры;

13)Не поддерживается extern;

3.Входные и выходные данные

Программа должна получать три параметра командной строки: имя входного файла с кодом программы, имя входного файла с переменными и имя файла для записи результата.

В файле с переменными переменные записаны в одну строку и могут быть разделены только одинарными пробелами.

В соответствии с правилами языка С переменные могут содержать буквы и нижнее подчеркивание.

Входные данные представляются в виде двух файлов с расширением .c и .txt.

Выходной файл должен быть текстовым файлом с расширением .txt. Он должен содержать кол-во строк равное кол-ву переменных. В каждой строчке вывод по каждой переменной с новой строки.

Примеры входных и выходных данных в приложение А.

4.Требование к надежности

В процессе работы программы не должно происходить ее аварийного завершения или зависания. В случае ошибки во входных данных, пользователь должен получать сообщения, перечисленные в таблице 1, после

чего программа должна корректно завершаться. Сообщения об ошибке выводятся в консоль, выходной файл при этом не создается.

Таблица 1 – Список сообщений об ошибках

Ситуация	Пример	Сообщение об ошибке
Указанный входной файл не существует, нет доступа к указанному файлу.		Неверно указан файл с входными данными. Возможно, файл не существует.
Невозможно создать указанный выходной файл.		Неверно указан файл для выходных данных. Возможно указанного расположения не существует или нет прав на запись.
Код программы содержит более 100 строк.		Код программы в принимаемом на вход файле должен содержать не более 100 строк.
Строки в коде программы содержат более 100 символов.		Строки в коде программы в принимаемом на вход файле должен содержать не более 100 символов.
Передан пустой файл.		Принимаемый на вход файл не может быть пустым.
В файле с переменными нет переменных.		В файле с переменными должны быть указаны переменные.

В файле с переменными указаны более 100 переменных.		В файле с переменными должны быть указаны не более 100 переменных.
В названиях переменных содержатся недопустимые символы: @, #, $, % и т.д в файле с переменными.	c@r #num	Названия переменных не должны содержать специальные символы.
В файле с переменными содержаться недопустимые разделители: . , / табы, множественные пробелы.	a, f. b	В файле с переменными присутствуют недопустимые разделители.
Каждая переменная в файле для переменных находится с новой строки.	a b c	Переменные в файле должны быть записаны в одну строку и разделены одинарными пробелами.
Имя переменной не может начинаться с числа.	1num 6res	Переменные в файле с переменными не должны начинаться с чисел.
Имя переменной не может совпадать с ключевыми словами языка С.	int char	Переменные в файле с переменными не должны совпадать с ключевыми словами языка С.
Имя переменной содержит буквы отличные от	hum rez	Имя переменной должна содержать только латинские

латинских.		буквы, цифры, нижние подчеркивания.
В коде программы присутствует #define.	#define WIDTH 80	В принимаемом на вход файле не может содержаться #define.
В коде программы присутствует typedef.	typedef struct X {};	В принимаемом на вход файле не может содержаться typedef.
В коде программы присутствует extern.	extern int i;	В принимаемом на вход файле не может содержаться extern.

5.Требования программной совместимости

Программа будет разработана на языке C++ с использованием среды разработки Visual Studio. Дополнительного программного обеспечения не требуется.

Входные файлы могут быть подготовлены в редакторе Блокнот.

Выходные файлы могут быть прочитаны с его помощью.

Приложение А

Входные и выходные данные

Таблица 2 – Примеры входных и выходных данных

code.c	var.txt	output.txt

#include <stdio.h>
int main(int mai)
{
int a = 3;
//int b;
int sum = 0;
sum = a + 3;
sum = sum * 2;
printf(" % \n", sum);
}
	a	a - объявлена
		b - не объявлена

#include <stdio.h>
int main(int argc)
{
int a = 3, b = 4, d;
d = MathFunc(a, b);
printf(\"%\n\", d);
}
int MathFunc(int A, int B)
{
int sum = 0;
sum = A + B;
sum = sum * 2;
	a sum	a - объявлена
		sum - объявлена

}		

struct str
{
int a;
int b;
}str1;
#include <stdio.h>
int main(int mai)
{
//int a;
int sum = 0;
sum = a + 3;
sum = sum * 2;
printf(" % \n", sum); }
	str1 b	str1 - объявлена
		b - не объявлена