ГУАП

КАФЕДРА № 14

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доц., канд. техн. наук |  | / |  | Е.П. Овсянников |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 11 |
| Синтаксический разбор текста |
| по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 1842 |  | / |  | А.Д. Аничкин |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2019

1. Цель работы

Закрепление знаний об основных конструкциях языка Си, закрепление навыков самостоятельной разработки алгоритмов и выбора инструментария для их реализации.

1. Задание

Задача 11.2

Вывести на экран содержимое файла с программой на языке С, выделяя в строке все целые константы.

1. Формализация задачи

Для решения поставленной задачи мы используем функцию GetStdHandle() для получения дескриптора окна консоли, с помощью которого, при помощи функции SetConsoleTextAttribute() мы будем менять цвет выходящего потока данных на консоль. Обработка файла будет вестись построчно. Целые константы будут выделены красным цветом на чёрном фоне.

1. Тестовые примеры

Исходные данные 1:  
#pragma warning(disable : 4996) // отключение ошибки c4996

#include <stdio.h>

int main(void)

{

float s; // сумма ряда

float a; // очередной элемент

int i; // номер элемента

int N; // количиство проходов

int z; // очередной элемент знаменателя

// ввод параметров ряда

printf("N ="); // вывод подсказки 'N ='

scanf("%d", &N); // запись значения с клавиатуры в нужном типе данных,

// где &(амперсанда) адрес

// начальное присвоение переменных

i = 0;

z = -2;

s = 0;

a = 1;

// цикл

while (i != N) // цикл с предусловием

{

z = z + 3; // увеличиваем дополнительное значение числителя

a = a \* ((float)(2 \* i + 1) / z); // рассчитываем очередной элемент

s = s + a; // составление суммы ряда

i = i + 1; // увеличиваем порядковый номер элемента

}

// вывод результата

printf("s = %21.20f\n", s); // %x,y(тип данных)

return 0;

}

Исходные данные 2:  
#pragma warning(disable : 4996) // отключение ошибки c4996

#include <stdio.h>

#include <conio.h> // библиотека обслуживающая '\_getch()' ???

#define YES 1

#define NO 0

int main(void)

{

// начальное присваивание

int c; // текущий символ из потока

int counter; // количество 'a'

int flag; // флажок, по которому мы

//определяем наличие буквы а в слове

// начальные присваивания

counter = 0;

flag = NO; // изначально обнуляем флажок

printf("Please, enter some words:\n");

// цикл чтоения символов из потока, связанного с клавиатурой

while ( (c = getchar()) != EOF )

{

if (c == 'a' || c == 'A') // выявляем наличие буквы а

{

flag = YES;

}

if (c == ' ' || c == '.' || c == '\n' || c == ',')

{

if (flag == YES) // проверка на наличие буквы а в слове

{

++counter; // увеличение счётчика на 1

flag = NO; // сброс флажка

}

}

}

printf("Number of words that contains letters 'a' = %d\n", counter);

printf("\nPress any button to close the window\n");

\_getch(); // ожидание ввода чего либо

return 0;

}

1. Листинг программы
2. #pragma warning(disable : 4996) // отключение ошибки c4996
3. #include <stdio.h>
4. #include <windows.h>
5. #include <locale.h>
6. #define STR 1024
7. void processing(char num[STR / 8]);
8. void main(void)
9. {
10. char \*locale = setlocale(LC\_ALL, "");
11. WORD foregroundColor0; // цвет слов
12. WORD textAttribute; // атрибут текста - цвет символа и фона
13. HANDLE hStdout = GetStdHandle(STD\_OUTPUT\_HANDLE);
14. foregroundColor0 = FOREGROUND\_INTENSITY | FOREGROUND\_BLUE | FOREGROUND\_GREEN;
15. textAttribute = foregroundColor0;
16. SetConsoleTextAttribute(hStdout, textAttribute);
17. int i; // счётчик
18. int j; // счётчик
19. char str[STR]; // массив строки
20. char num[STR / 8]; // массив целочисленной константы
21. char \*str\_ptr = str;
22. FILE \*frd;
23. printf("Please enter address of file: ");
24. gets(str); // получаем путь к файлу
25. if (\*str\_ptr == '"')
26. { // удаление ковычек
27. i = 0;
28. while (\*(str\_ptr + i) != '\0')
29. {
30. (\*(str\_ptr + i + 1) == '"') ? (\*(str\_ptr + i) = '\0') : (\*(str\_ptr + i) = \*(str\_ptr + i + 1));
31. ++i;
32. }
33. }
34. printf("Path of file: %s\n", str); // отладка
35. frd = fopen((str), "rt");
36. if (frd == NULL)
37. return; // chk open file
38. while (!feof(frd)) // for EOF
39. {
40. fgets(str, STR, frd); // get file string
41. str\_ptr = str;
42. i = 0;
43. while (\*(str\_ptr + i) != '\0')
44. {
45. if (\*(str\_ptr + i) == '/' && \*(str\_ptr + i + 1) == '/')
46. { // защита от //
47. while (\*(str\_ptr + i) != '\0')
48. {
49. printf("%c", \*(str\_ptr + i));
50. ++i;
51. }
52. break;
53. }
54. if (\*(str\_ptr + i) == '-' || (\*(str\_ptr + i) >= '0' && \*(str\_ptr + i) <= '9'))
55. {
56. j = 0;
57. while (\*(str\_ptr + i) != '\0' && \*(str\_ptr + i) != ';' && \*(str\_ptr + i) != ' ' && \*(str\_ptr + i) != ',' && \*(str\_ptr + i) != '}' && \*(str\_ptr + i) != ']' && \*(str\_ptr + i) != '\n')
58. {
59. num[j] = \*(str\_ptr + i);
60. ++i;
61. ++j;
62. }
63. --i;
64. num[j] = '\0';
65. processing(num);
66. }
67. else
68. printf("%c", \*(str\_ptr + i));
69. ++i;
70. }
71. }
72. fclose(frd);
73. return 0;
74. }
75. void processing(char num[STR / 8])
76. {
77. WORD foregroundColor0; // цвет обычный
78. WORD foregroundColor1; // цвет необычный
79. WORD textAttribute; // атрибут текста - цвет символа и фона
80. HANDLE hStdout = GetStdHandle(STD\_OUTPUT\_HANDLE);
81. foregroundColor0 = FOREGROUND\_INTENSITY | FOREGROUND\_BLUE | FOREGROUND\_GREEN;
82. foregroundColor1 = FOREGROUND\_INTENSITY | FOREGROUND\_RED;
83. int j = 0;
84. int value = strlen(num);
85. while (num[j] == '-' || (num[j] >= '0'&& num[j] <= '9'))
86. ++j;
87. if (value == j)
88. {
89. textAttribute = foregroundColor1;
90. SetConsoleTextAttribute(hStdout,textAttribute);
91. printf("%s", num);
92. textAttribute = foregroundColor0;
93. SetConsoleTextAttribute(hStdout, textAttribute);
94. }
95. else
96. printf("%s", num);
97. return;
98. }
99. Вывод; д-во правильности работы

 Программа работает, что подтверждают тестовые примеры и снимки экрана



