**G1**

Три раза

В файле .txt дана строка. Вывести ее в файл .txt три раза через запятую и показать количество символов в ней.

Данные на входе: Строка из английских букв и пробелов. Не более 100 символов. В конце могут быть незначащие переносы строк.

Данные на выходе: Исходная строка 3 раза подряд, через запятую пробел и количество символов в ней.

Пример

Данные на входе: aab

Данные на выходе: aab, aab, aab 3

| **#include <stdio.h>**    **const int line\_width = 256;**    **int main(void)**  **{**  **char \* input\_fn = "input.txt";**  **char \* output\_fn = "output.txt";**  **char line[line\_width];**  **char c;**  **FILE \*fp;**  **if((fp = fopen(input\_fn, "r")) == NULL)**  **{**  **perror("Error occured while opening input file!");**  **return 1;**  **}**  **int count = 0;**  **while(((c = getc(fp)) != EOF) && (c != '\n'))**  **line[count++] = c;**  **line[count] = '\0';**    **fclose(fp);**  **if((fp = fopen(output\_fn, "w")) == NULL)**  **{**  **perror("Error occured while opening output file!");**  **return 1;**  **}**  **for (int i = 0; i < 3; i++)**  **{**  **if (i)**  **fprintf(fp, ", ");**  **fprintf(fp, "%s", line);**  **}**  **fprintf(fp, " %d", count);**  **fclose(fp);**  **return 0;**  **}** |
| --- |

**G2 Демо 1**

Строка и цифры

Считать число N из файла input.txt. Сформировать строку из N символов. N четное число, не превосходящее 26. На четных позициях должны находится четные цифры в порядке возрастания, кроме 0, на нечетных позициях - заглавные буквы в порядке следования в английском алфавите. Результат записать в файл output.txt

Данные на входе: Четное N

Данные на выходе: Строка из английских букв и цифр

Пример №1

Данные на входе: 10

Данные на выходе: A2B4C6D8E2

Пример №2

Данные на входе: 16

Данные на выходе: A2B4C6D8E2F4G6H8

| **#include <stdio.h>**  **#include <string.h>**    **int main(){**  **int i, n;**    **FILE \*f\_in, \*f\_out;**  **char name\_in[] = "input.txt";**  **char name\_out[] = "output.txt";**  **f\_in = fopen(name\_in, "r");**  **f\_out = fopen(name\_out, "w");**    **fscanf(f\_in, "%d", &n);**    **char str[n + 2];**  **char c = 'A';**  **char c\_num = '2';**  **for (i = 0; i < n; ++i)**  **{**  **if (i%2==0)**  **str[i] = c++;**  **else**  **{**  **str[i] = c\_num;**  **if (c\_num == '8')**  **c\_num = '2';**  **else**  **c\_num += 2;**  **}**  **}**    **str[i] = '\0';**  **fprintf(f\_out, "%s", str);**  **fclose(f\_in);**  **fclose(f\_out);**  **return 0;**  **}** |
| --- |

**G3 Практ 1**

Последний номер символа

В файле .txt дана строка из 1000 символов. Показать номера символов, совпадающих с последним символом строки. Результат записать в файл .txt

Данные на входе: Строка не более 1000 символов

Данные на выходе: Целые числа через пробел - номера символа, который совпадает с последним символом строки.

Пример

Данные на входе: aabbcdb

Данные на выходе: 2 3

**G4 ДЗ 1**

По одному разу

В файле .txt даны два слова не более 100 символов каждое, разделенные одним пробелом. Найдите только те символы слов, которые встречаются в обоих словах только один раз. Напечатайте их через пробел в файл .txt в лексикографическом порядке.

Данные на входе: Два слова из маленьких английских букв через пробел. Длинна каждого слова не больше 100 символов.

Данные на выходе: Маленькие английские буквы через пробел.

Пример №1

Данные на входе: hello world

Данные на выходе: o

Данные на входе: aabcd bcef

Данные на выходе: b c

**G5 ДЗ 2**

Заменить a на b

В файле .txt дана символьная строка не более 1000 символов. Необходимо заменить все буквы "а" на буквы "b" и наоборот, как заглавные, так и строчные. Результат записать в .txt.

Данные на входе: Строка из маленьких и больших английских букв, знаков препинания и пробелов.

Данные на выходе: Строка из маленьких и больших английских букв, знаков препинания и пробелов.

Пример

Данные на входе: aabbccddABCD

Данные на выходе: bbaaccddBACD

**G6 Демо 2**

Проверка на палиндром

В файле .txt символьная строка не более 1000 символов. Необходимо проверить, является ли она палиндромом (палиндром читается одинаково в обоих направлениях). Реализовать логическую функцию is\_palindrom(str) и записать ответ в файл .txt.

Данные на входе: Строка из заглавных английских букв

Данные на выходе: YES или NO

Пример №1

Данные на входе: ABCDEF

Данные на выходе: NO

Пример №2

Данные на входе: KAZAK

Данные на выходе: YES

| **#include <stdio.h>**  **#include <string.h>**    **const int line\_width = 1024;**    **int is\_palindrome(char \* line)**  **{**  **int len = strlen(line);**  **for (int i = 0; i < len / 2; i++)**  **if (line[i] != line[len - i - 1])**  **return 0;**  **return 1;**  **}**    **int main(void)**  **{**  **char \* input\_fn = "input.txt";**  **char \* output\_fn = "output.txt";**  **char line[line\_width];**  **FILE \*fp;**  **fp = fopen(input\_fn, "r");**  **fscanf(fp, "%[^\n]", line);**  **fclose(fp);**  **fp = fopen(output\_fn, "w");**  **fprintf(fp, "%s", is\_palindrome(line) ? "YES" : "NO");**  **fclose(fp);**  **return 0;**  **}** |
| --- |

**G7 Практ 2**

Количество букв

В файле .txt считать символьную строку, не более 10 000 символов. Посчитать количество строчных (маленьких) и прописных (больших) букв в введенной строке. Учитывать только английские буквы. Результат записать в файл .txt.

Данные на входе: Строка состоящая из английских букв, цифр, пробелов и знаков препинания.

Данные на выходе: Два целых числа. Количество строчных букв и количество заглавных букв.

Пример №1

Данные на входе: aabbAB

Данные на выходе: 4 2

Пример №2

Данные на входе: HELLO WORLD

Данные на выходе: 0 10

**G8 ДЗ 3**

Числа в массив

В файле .txt дана строка, не более 1000 символов, содержащая буквы, целые числа и иные символы. Требуется все числа, которые встречаются в строке, поместить в отдельный целочисленный массив. Например, если дана строка "data 48 call 9 read13 blank0a", то в массиве числа 48, 9, 13 и 0. Вывести массив по возрастанию в файл .txt.

Данные на входе: Строка из английских букв, цифр и знаков препинания

Данные на выходе: Последовательность целых чисел отсортированная по возрастанию

Пример

Данные на входе: data 48 call 9 read13 blank0a

Данные на выходе: 0 9 13 48

**G9 ДЗ 4**

Удалить повторяющиеся символы

В файле .txt строка из меленьких и больших английских букв, знаков препинания и пробелов. Требуется удалить из нее повторяющиеся символы и все пробелы. Результат записать в файл .txt.

Данные на входе: Строка из меленьких и больших английских букв, знаков препинания и пробелов. Размер строки не более 1000 сивмолов.

Данные на выходе: Строка из меленьких и больших английских букв.

Пример

Данные на входе: abc cde def

Данные на выходе: abcdef

**G10**

Самое длинное слово

В файле .txt дана строка слов, разделенных пробелами. Найти самое длинное слово и вывести его в файл .txt. Случай, когда самых длинных слов может быть несколько, не обрабатывать.

Данные на входе: Строка из английских букв и пробелов. Не более 1000 символов.

Данные на выходе: Одно слово из английских букв.

Пример

Данные на входе: Hello beautiful world

Данные на выходе: beautiful

**G11**

Заканчивается на a

В файле .txt дано предложение. Необходимо определить, сколько слов заканчиваются на букву 'а'. Ответ записать в файл .txt.

Данные на входе: Строка из английских букв и пробелов не более 1000 символов.

Данные на выходе: Одно целое число

Пример

Данные на входе: Mama mila ramu

Данные на выходе: 2

**G12**

Разобрать на слова

В файле .txt дано предложение требуется разобрать его на отдельные слова. Напечатать каждое слово на отдельной строке в файл .txt.

Данные на входе: Одна строка из английских букв и пробелов не более 1000 символов.

Данные на выходе: Каждое слово на отдельной строке

Пример

Данные на входе: Mama mila ramu

Данные на выходе: Mama mila ramu

**G13**

Изменить расширение

В файле .txt записан полный адрес файла (возможно, без расширения). Необходимо изменить его расширение на ".html" и записать результат в файл .txt.

Данные на входе: Строка состоящая из символов: a-z, A-Z, 0-9, /

Данные на выходе: Исходная строка с измененным расширением.

Пример №1

Данные на входе: /DOC.TXT/qqq

Данные на выходе: /DOC.TXT/qqq.html

Пример №2

Данные на входе: /DOC.TXT/qqq.com

Данные на выходе: /DOC.TXT/qqq.html

**G14**

Hello name

В файле .txt в одной строке фамилию, имя и отчество. Сформировать файл приветствие .txt, где останутся имя и фамилия

Данные на входе: Строка состоящая из английских букв и пробелов не более 100 символов. Формат: Фамилия Имя Отчество

Данные на выходе: Строка состоящая из английских букв и пробелов

Пример

Данные на входе: Pupkin Vasiliy Ivanovich

Данные на выходе: Hello, Vasiliy Pupkin!

**G15 Демо 3**

Cao заменить на Ling

В файле .txt дано предложение. Необходимо заменить все имена «Cao» на «Ling» и записать результат в файл .txt.

Данные на входе: Строка из английский букв, знаков препинания и пробелов. Не более 1000 символов.

Данные на выходе: Строка из английский букв, знаков препинания и пробелов.

Пример

Данные на входе: Cao, Cao, Cao and Cao!!!

Данные на выходе: Ling, Ling, Ling and Ling!!!

| **#include <stdio.h>**  **#include <string.h>**    **#define SIZE 10001**    **void input(char \*strIn)**  **{**  **FILE \*in;**    **in = fopen("input.txt", "r");**    **while(fscanf(in, "%[^\n]", strIn) == 1);**    **fclose(in);**  **}**    **void changeName(char \*strExt, char \*buffer)**  **{**  **int count = 0;**    **for(int i = 0; i < strlen(strExt); i++)**  **{**  **if(!memcmp(strExt+i,"Cao",3))**  **{**  **strcat(buffer, "Ling");**  **count += 4;**  **i += 3;**  **}**  **buffer[count++] = strExt[i];**  **}**  **}**  **void output(char \*strOut)**  **{**  **FILE \*out;**  **out = fopen("output.txt", "w");**  **fprintf(out, "%s", strOut);**  **fclose(out);**  **}**    **int main(int argc, char \*\*argv)**  **{**  **char str[SIZE]={0}, strFinal[SIZE]={0};**  **input(str);**  **changeName(str, strFinal);**  **output(strFinal);**  **return 0;**  **}** |
| --- |

**G16 Практ 3**

Ling заменить на Cao

В файле .txt дано предложение. Необходимо заменить все имена «Ling» на «Cao» и результат записать в файл .txt.

Данные на входе: Строка из английских букв, пробелов и знаков препинания. Не более 1000 символов.

Данные на выходе: Строка из английских букв, пробелов и знаков препинания.

Пример

Данные на входе: Ling, Ling, Ling and Ling!!!

Данные на выходе: Cao, Cao, Cao and Cao!!!

**G17**

Пары соседних символов

В файле .txt записаны символы. Необходимо разработать функцию, которая меняет местами пары соседних символов не обращая внимание на символы пробел. Если количество символов нечетно (пробелы не считаем), то последний символ не меняем. Результат записать в файл .txt.

Данные на входе: Строка из английских букв, пробелов и знаков препинания. Не более 1000 символов.

Данные на выходе: Строка из английских букв, пробелов и знаков препинания.

Пример №1

Данные на входе: Hello world!

Данные на выходе: eHllw orodl!

Пример №2

Данные на входе: abc def

Данные на выходе: bad cfe

**G18**

Удалить пробелы из текста

В файле .txt необходимо удалить все лишние пробелы (в начале предложения и сдвоенные пробелы). Для решения задачи разработать функцию. Результат записать в .txt.

Данные на входе: Строка из английских букв, знаков препинания и пробелов. Не более 1000 символов.

Данные на выходе: Строка из английских букв, знаков препинания и пробелов.

Пример

Данные на входе: Hello world!

Данные на выходе: Hello world!

**G19 Демо 2**

Составить палиндром

Разработать функцию дана строка из маленьких английских букв. Составить из символов палиндром максимальной длинны. При составлении палиндрома буквы в палиндроме должны быть расположены в лексикографическом порядке. Записать ответ в файл .txt.

Данные на входе: Строка из маленьких английских букв. Не более 1000 символов.

Данные на выходе: Строка из маленьких английских букв.

Пример №1

Данные на входе: kazaki

Данные на выходе: akika

Пример №2

Данные на входе: knak

Данные на выходе: kak

Пример №3

Данные на входе: zzzzbbaaa

Данные на выходе: abzzazzba

| **#include <stdio.h>**  **#define ALPHABET 26**  **int main()**  **{**  **FILE \*f\_in = fopen("input.txt", "r");**  **FILE \*f\_out = fopen("output.txt", "w");**    **char str[1002];**  **int odd;**  **int a[ALPHABET] = {0};**    **fgets(str, 1002, f\_in);**    **for(int i=0; str[i] != '\n' && str[i]; ++i)**  **a[str[i] - 'a']++;**    **for(int i = 0; i < ALPHABET; ++i)**  **if (a[i] % 2 != 0)**  **{**  **odd = i;**  **break;**  **}**    **for(int i = 0; i < ALPHABET; ++i)**  **if (a[i] > 0)**  **for (int j = 0; j < a[i]/2; ++j)**  **fprintf(f\_out, "%c", i + 'a');**    **fprintf(f\_out, "%c", odd + 'a');**    **for(int i = ALPHABET-1; i >= 0; --i)**  **if (a[i] > 0)**  **for (int j = 0; j < a[i]/2; ++j)**  **fprintf(f\_out, "%c", i + 'a');**  **fclose(f\_in);**  **fclose(f\_out);**  **return 0;**  **}** |
| --- |

| **#include <stdio.h>**  **#include <string.h>**    **const int line\_width = 1024;**    **void bubble\_sort(int size, char\* array)**  **{**  **char buf;**  **for (int i = 0; i < size - 1; i++)**  **for (int j = i + 1; j < size; j++)**  **if (array[i] > array[j])**  **{**  **buf = array[i];**  **array[i] = array[j];**  **array[j] = buf;**  **}**  **}**    **void make\_palindrome(int size, char\* line)**  **{**  **char letters[size][2];**  **int letters\_count = 0;**  **int idx = 0;**  **bubble\_sort(strlen(line), line);**  **for (int i = 0; i < size; i++)**  **{**  **idx = -1;**  **for (int j = 0; j < letters\_count; j++)**  **if (line[i] == letters[j][0])**  **{**  **idx = j;**  **letters[idx][1]++;**  **break;**  **}**  **if (idx < 0)**  **{**  **idx = letters\_count++;**  **letters[idx][0] = line[i];**  **letters[idx][1] = 1;**  **}**  **}**  **int p\_length = 0;**  **int l\_length, m\_p = -1;**  **for (int i = 0; i < letters\_count; i++)**  **{**  **while (letters[i][1] && (letters[i][1] / 2 > 0))**  **{**  **line[p\_length++] = letters[i][0];**  **letters[i][1] -= 2;**  **}**  **if ((m\_p < 0) && (letters[i][1] % 2 == 1))**  **m\_p = i;**  **}**  **l\_length = p\_length;**  **if (m\_p >= 0)**  **line[p\_length++] = letters[m\_p][0];**  **for (int i = l\_length - 1; i >= 0; i--)**  **line[p\_length++] = line[i];**  **line[p\_length] = '\0';**  **}**    **int main(void)**  **{**  **char \* input\_fn = "input.txt";**  **char \* output\_fn = "output.txt";**  **char line[line\_width];**  **FILE \*fp;**  **fp = fopen(input\_fn, "r");**  **fscanf(fp, "%[^\n]", line);**  **fclose(fp);**  **make\_palindrome(strlen(line), line);**  **fp = fopen(output\_fn, "w");**  **fprintf(fp, "%s", line);**  **fclose(fp);**  **return 0;**  **}** |
| --- |

**G20 Практ 2**

Проверка на палиндром

Считать предложение из файла .txt и определить можно ли из английских букв предложения записанного в файле получить одно слово - палиндром. Ответ напечатать на стандартный поток вывода. Требуется реализовать логическую функцию и применить ее.

is\_palindrom(string)

Данные на входе: Строка из маленьких английских букв и пробелов. Не более 1000 символов.

Данные на выходе: YES или NO

Пример №1

Данные на входе: kak za

Данные на выходе: YES

Пример №2

Данные на входе: abc

Данные на выходе: NO

Пример №3

Данные на входе: car race

Данные на выходе: YES

**G21 Демо 2+**

Равносторонние треугольники

Cia очень любит равносторонние треугольники. Он все время выкладывает их из камней. Ling дал Cia камни и попросил его составить такой треугольник если это возможно. Помогите Cia сложить треугольник используя все камни или напечатайте слов NO, если это невозможно. Пример треугольников, которые составляет Cia:

\*

\* \*

\*

\* \*

\* \* \*

\*

\* \*

\* \* \*

\* \* \* \*

Во входном файле .txt записано некоторое количество символов \* (камней). Необходимо построить равносторонний треугольник используя все символы \* и символ пробел, записать ответ в выходной файл .txt. Между соседними символами \* строго один пробел. Если треугольник невозможно составить, используя все камни, то необходимо записать единственное слово NO в файл .txt.

Данные на входе: Текстовый файл состоит из различных символов и символов \*

Данные на выходе: Равносторонний треугольник составленный из символов \*, символов пробела и символов переноса строк или слово NO

Пример №1

Данные на входе: \*\*\*

Данные на выходе: \*

\* \*

Пример №2

Данные на входе: \*\* hello world \*\*\* \*

Данные на выходе: \*

\* \*

\* \* \*

Пример №3

Данные на входе: \*\*\*\*

Данные на выходе: NO

| **#include <stdio.h>**    **const int size = 1024;**    **int main(void)**  **{**  **char \* input\_fn = "input.txt";**  **char \* output\_fn = "output.txt";**  **FILE \*fp;**  **char lines[size][size];**    **int count = 0;**  **fp = fopen(input\_fn, "r");**  **char c;**  **while ((c = getc(fp)) != EOF)**  **if (c == '\*') count++;//считаем камни**  **fclose(fp);**  **int k, row\_count = 0;**  **while (count > 0)//пока камни не кончатся**  **{**  **row\_count++;//счетчик столбцов**  **if (count < row\_count)**  **break;**    **for (int j = 0; j < row\_count; j++)**  **{**  **k = row\_count - j - 1;**  **for (int i = 0; i < k; i++)**  **lines[j][i] = ' ';**  **for (int i = 0; i < j + 1; i++)**  **{**  **lines[j][i \* 2 + k] = '\*';**  **lines[j][i \* 2 + k + 1] = ' ';**  **}**  **lines[j][j \* 2 + k + 1] = '\0';**  **}**    **count -= row\_count;**  **}**  **fp = fopen(output\_fn, "w");**  **if (!count && row\_count)**  **for (int i = 0; i < row\_count; i++)**  **fprintf(fp, (i) ? "\n%s" : "%s", lines[i]);**  **else**  **fprintf(fp, "%s", "NO");**  **fclose(fp);**  **return 0;**  **}** |
| --- |

**G22**

Soundex

Известный алгоритм Soundex (https://ru.wikipedia.org/wiki/Soundex) определяет, похожи ли два английских слова по звучанию. На вход он принимает слово и заменяет его на некоторый четырёхсимвольный код. Если коды двух слов совпадают, то слова, как правило, звучат похоже.

Вам требуется реализовать этот алгоритм. Он работает так:

Первая буква слова сохраняется.

В остальной части слова:

буквы, обозначающие, как правило, гласные звуки: a, e, h, i, o, u, w и y — отбрасываются; оставшиеся буквы (согласные) заменяются на цифры от 1 до 6, причём похожим по звучанию буквам соответствуют одинаковые цифры:

1: b, f, p, v

2: c, g, j, k, q, s, x, z

3: d, t

4: l

5: m, n

6: r

Любая последовательность одинаковых цифр сокращается до одной такой цифры. Итоговая строка обрезается до первых четырёх символов. Если длина строки меньше требуемой, недостающие символы заменяются знаком 0.

Примеры:

аmmonium → ammnm → a5555 → a5 → a500

implementation → implmnttn → i51455335 → i514535 → i514

Данные на входе: Во входном файле .txt находится одно непустое слово, записанное строчными латинскими буквами. Длина слова не превосходит 20 символов.

Данные на выходе: Напечатайте четырёхбуквенный код, соответствующий слову в файл .txt.

Пример №1

Данные на входе: ammonium

Данные на выходе: a500

Пример №2

Данные на входе: implementation

Данные на выходе: i514