//problema 3

#include <iostream>

#include <fstream>//Para abrir un fichero para lectura, debe crear un objeto ifstream que se usará como cin.

#include <string>//cadenas de texto

#include <vector>

#include <iomanip>//para manejar formatos, posicionar el texto donde quieras(centrado) setw

using namespace std;

int main() {

string archivoNombre = "Ejemplo1.txt";

ifstream miarchivo(archivoNombre);

string parrafo;

vector<string> VectorParrafo;

bool pruebas=0;

if(miarchivo.is\_open()){

while(getline(miarchivo,parrafo)){//getline lee las lineas del texto

VectorParrafo.push\_back(parrafo);

}

miarchivo.close();

}

string linea;

string lineaPart;

int ancho=72;

int contLetras=0;

int posUltimoE;

for(int i=0; i<VectorParrafo.size(); i++){

linea = VectorParrafo[i];

//aquí empieza

while(linea.length()>ancho){

lineaPart = linea.substr(0,ancho);

if (pruebas==1)

cout << "P1 " << lineaPart << endl;

//si la ultima es espacio

if(lineaPart[ancho-1]==' '){

//modificar linea

linea = linea.substr(ancho);

if (pruebas==1){

cout << "Entrada 1" << endl;

}

}else{

//si no es espacio, ver si es la ultima letra de una palabra

if(linea[ancho]==' '){

if (pruebas==1){

cout << "Entrada 2" << endl;

}

//es la ultima palabra, no hay problema entonces

//modificar linea

linea = linea.substr(ancho+1);

lineaPart = lineaPart+" ";

}else{

if (pruebas==1){

cout << "Entrada 3" << endl;

}

//tiene una palabra entrecortada, contar cuantas letras se eliminaran y agregaran como espacios

posUltimoE = ancho-1;

for(int m=ancho-1;m>=0;m--){

if(lineaPart[m]==' '){

break;

}else{

posUltimoE--;

}

}

contLetras = ancho-posUltimoE-1;

//modificar linea

linea = linea.substr(ancho-contLetras);

//eliminar estas letras

lineaPart = lineaPart.substr(0,ancho-contLetras);

}

}

//Afinamiento

//Elimina espacios adelante y atras

char letra=lineaPart[0];

while(letra==' '){

lineaPart = lineaPart.substr(1);

letra = lineaPart[0];

}

lineaPart[lineaPart.length()-1]='\_';

if (pruebas==1){

cout << lineaPart << endl;

for(int k=0; k<lineaPart.length();k++)

cout << k%10;

cout << endl;

}

int longitud;

for(int k=lineaPart.length()-1;k>=0; k--){

if(lineaPart[k]=='\_'){

longitud=k;break;

}

}

int cantEspaciosFaltantes = ancho-longitud;

while(cantEspaciosFaltantes!=0){

for(int k=0; k<lineaPart.length(); k++){

if(lineaPart[k]==' '){

lineaPart.insert(k," ");

cantEspaciosFaltantes--;

if(cantEspaciosFaltantes==0) break;

if(cantEspaciosFaltantes<0) cantEspaciosFaltantes=0;

}

while(lineaPart[k]==' ' & k<lineaPart.length()){

k++;

}

}

if(cantEspaciosFaltantes==0) break;

}

lineaPart[lineaPart.length()-1]=' ';

cout << lineaPart << endl;

if (pruebas==1) {

for (int k = 0; k < lineaPart.length(); k++)

cout << k % 10;

cout << endl;

}

if(linea.length()<=ancho){

cout << linea << endl << endl;

}

}

}

}