#include <iostream>

using namespace std;

// C- String 2

// Task 1

// 1. Sətirdəki bütün saitləri sayan program yazın. ( A E İ O U).

// This function returns the number of vowels in a string

int vowelCounter(char\*& str)

{

int l = strlen(str);

int counter = 0;

for (int x = 0; x < l; x++)

{

if (str[x] == 'a' || str[x] == 'e' || str[x] == 'i'

|| str[x] == 'o' || str[x] == 'u' || str[x] == 'A'

|| str[x] == 'E' || str[x] == 'I' || str[x] == 'O'

|| str[x] == 'U')

counter++;

}

return counter;

}

// Task 2

// 2. Sətirdəki bütün samitləri sayan program yazın.

// This function returns the number of consonants in a string

int consonantCounter(char\*& str)

{

int l = strlen(str);

int counter = 0;

// Yoxlaması daha uzun olar deyə yazmadım (Hamısının true olub-olduğunu yoxlayır)

// (str[x] != 'a' && str[x] != 'e' && str[x] != 'i'

// && str[x] != 'o' && str[x] != 'u' && str[x] != 'A'

// && str[x] != 'E' && str[x] != 'I' && str[x] != 'O'

// && str[x] != 'U')

for (int x = 0; x < l; x++)

{

// (Biri true olanda counter artır)

if (str[x] == 'q' || str[x] == 'w' || str[x] == 'r'

|| str[x] == 't' || str[x] == 'y' || str[x] == 'p'

|| str[x] == 's' || str[x] == 'd' || str[x] == 'f'

|| str[x] == 'g' || str[x] == 'h' || str[x] == 'j'

|| str[x] == 'k' || str[x] == 'l' || str[x] == 'z'

|| str[x] == 'x' || str[x] == 'c' || str[x] == 'v'

|| str[x] == 'b' || str[x] == 'n' || str[x] == 'm'

|| str[x] == 'Q' || str[x] == 'W' || str[x] == 'R'

|| str[x] == 'T' || str[x] == 'Y' || str[x] == 'P'

|| str[x] == 'S' || str[x] == 'D' || str[x] == 'F'

|| str[x] == 'G' || str[x] == 'H' || str[x] == 'J'

|| str[x] == 'K' || str[x] == 'L' || str[x] == 'Z'

|| str[x] == 'X' || str[x] == 'C' || str[x] == 'V'

|| str[x] == 'B' || str[x] == 'N' || str[x] == 'M')

counter++;

}

return counter;

}

// Task 3

// 3. Sətri PigLatin edən funksiya yazın.

// This function returns the Pig Latin form of the word

char\* pigLatin(char\* str)

{

int l = strlen(str);

auto\* newChar = new char[l+2] {};

char firstLetter = str[0];

for (int x = 1; x < l; x++)

{

newChar[x - 1] = str[x];

}

newChar[l-1] = firstLetter;

newChar[l] = 'a';

newChar[l + 1] = 'y';

newChar[l + 2] = '\0';

return newChar;

}

// Task 4

// 4. Funksiya sətirdəki boşluqları müəyyən simvolla

// əvəzləsin. (Simvolu istifadəçi funksiyaya göndərir)

// This function replaces spaces with the given symbol

char\* replaceSpaces(char\* str, char symbol)

{

int l = strlen(str);

char\* newChar = new char[l] {};

for (int x = 0; x < l; x++)

{

if (str[x] == ' ')

newChar[x] = symbol;

else

newChar[x] = str[x];

}

newChar[l] = '\0';

return newChar;

}

// Task 5

// 5. Sətir str ​sətir str2-nin ​suffixi olub olmadığını yoxlayan

// funksiya yazın.

// This function checks whether str is a suffix of str2

int isSuffix(char\* str1, char\* str2)

{

int l = strlen(str1);

int l2 = strlen(str2);

bool isSuffix = true;

for (int x = l2 - l, x2 = 0; x < l2; x++, x2++)

{

if (str1[x2] != str2[x])

{

isSuffix = false;

break;

}

}

if (isSuffix == true)

return 1;

return 0;

}

// Task 6

// 6. Sətir str ​sətir str2-nin ​prefixi olub olmadığını tapan

// funksiya yazın.

// This function checks whether str is a prefix of str2

int isPrefix(char\* str1, char\* str2)

{

int l = strlen(str1);

int l2 = strlen(str2);

bool isPrefix = true;

for (int x = 0; x < l; x++)

{

if (str1[x] != str2[x])

{

isPrefix = false;

break;

}

}

if (isPrefix == true)

return 1;

return 0;

}

// Task 7

// 7. Sətirin polindrom olub olmadığını yoxlayan funskiya

// yazın.

// This function checks whether the string is a palindrome

int isPalindrome(char\* str)

{

int l = strlen(str);

bool isPolyndrome = true;

for (int x = 0; x < l; x++)

{

if (str[x] != str[l - x - 1])

{

isPolyndrome = false;

break;

}

}

if (isPolyndrome == true)

return 1;

return 0;

}

// Task 8

// 8. Cmlədə nə qədər hərf nə qədər rəqəm nə qədər

// punktuasiya olduğunu hesablayan funksiya yazın.

// This function returns the number of numerals in the string

int numberCounter(char\* str)

{

int l = strlen(str);

int counter = 0;

for (int x = 0; x < l; x++)

{

if (str[x] >= 48 && str[x] <= 57)

counter++;

}

return counter;

}

// This function returns the number of letters in the string

int letterCounter(char\* str)

{

int l = strlen(str);

int counter = 0;

for (int y = 0; y < l; y++)

{

if (str[y] >= 65 && str[y] <= 90

|| str[y] >= 97 && str[y] <= 122)

counter++;

}

return counter;

}

// This function returns the number of punctuation marks in the string

int punctuationCounter(char\* str)

{

int l = strlen(str);

int counter = 0;

for (int z = 0; z < l; z++)

{

if (str[z] >= 33 && str[z] <= 47

|| str[z] >= 58 && str[z] <= 64

|| str[z] >= 91 && str[z] <= 96

|| str[z] >= 123 && str[z] <= 126)

counter++;

}

return counter;

}

// Task 9

// 9. -The​- sözünün cümlədə neçə dəfə olduğunu tapan

// funksiya yazın.

// This function returns the number of -the-s in the string

int theCounter(char\* str)

{

int l = strlen(str);

int counter = 0;

for (int x = 0; x < l - 2; x++)

{

if (str[x] == 't' && str[x+1] == 'h' && str[x+2] == 'e')

counter++;

}

return counter;

}

// Task 10

// 10. Sətirdəki hərflərdən başqa bütün simvolları və

// rəqəmləri silən funksiya yazmaq.

// This function returns the number of spaces in the string

int spaceCounter(char\* str)

{

int l = strlen(str);

int counter = 0;

for (int y = 0; y < l; y++)

{

if (str[y] == 32)

counter++;

}

return counter;

}

// This function deletes the numbers and punctuation marks in the string

char\* deleteNumberAndPunct(char\* str)

{

int l = strlen(str);

int a = letterCounter(str);

int b = spaceCounter(str);

int l2 = a + b;

char\* newArray = new char[l2] {};

int index = 0;

for (int x = 0; x < l; x++)

{

if (str[x] >= 65 && str[x] <= 90 || str[x] >= 97 && str[x] <= 122 || str[x] == 32)

{

newArray[index] = str[x];

index++;

}

}

newArray[l2] = '\0';

return newArray;

}

// Task 11

// 11. Cümlədə ən çox hansı hərfin işləndiyini tapan

// funksiya yazın.

// This function returns the string in the lower case

char\* mystrlow(char\* str1)

{

int l = strlen(str1);

for (int x = 0; x < l; x++)

{

if ((int)str1[x] >= 65 && (int)str1[x] <= 90)

str1[x] = char(int(str1[x]) + 32);

}

return str1;

}

// This function returns the copy of a string

void mystrcpy(char\*& str1, const char\* str2)

{

int l = strlen(str2);

str1 = new char[l + 1];

for (int x = 0; x < l; x++)

{

str1[x] = str2[x];

}

str1[l] = '\0';

}

// This function finds which letter is used the most in a string

char mostUsedLetter(char\* str)

{

int l = strlen(str);

int counter = 0;

int counter2 = 0;

char\* newChar = new char[l]{};

mystrcpy(newChar, str);

newChar = mystrlow(newChar);

char mostUsedLetter = newChar[0];

for (int x = 0; x < l; x++)

{

counter2 = 0;

for (int y = 0; y < l; y++)

{

if (newChar[x] == newChar[y])

counter2++;

}

if (counter2 > counter)

{

mostUsedLetter = newChar[x];

counter = counter2;

}

}

newChar = nullptr;

return mostUsedLetter;

}

// Task 12

// 12. Cümlədəki ən uzun sözü tapan funksiya yazın.

// This function finds the longest word in a sentence

void longestWord(char\* str)

{

int l = strlen(str);

int index = 0;

int wordLength = 0;

int longestWordLength = 0;

int longestWordIndex = 0;

int counter = 0;

int C = 2;

cout << "\n The length of all the words in the sentece : " << endl;

cout << "\n 1) ";

for (int x = 0; x <= l; x++)

{

if (str[x] == ' ' || str[x] == '\0')

{

cout << " - " << wordLength << ".\n " << C << ") ";

if (longestWordLength < wordLength)

{

longestWordLength = wordLength;

longestWordIndex = counter - wordLength;

}

wordLength = 0;

counter++;

C++;

}

else

{

cout << str[x];

wordLength++;

counter++;

}

if (x == l)

cout << "\b\b\b\b " << endl;

}

cout << " The Length Of The Longest Word : " << longestWordLength << "." << endl;

cout << " The Index Of The Longest Word : " << longestWordIndex << "." << endl;

cout << "\n The First Longest Word : ";

for (int y = longestWordIndex; y < longestWordIndex + longestWordLength; y++)

{

cout << str[y];

}

cout << endl;

}

// Task 13

// 13. Cümlədəki ən kiçik sözü tapan funksiya tapın

// This function finds the shortest word in a sentence

void shortestWord(char\* str)

{

int l = strlen(str);

int index = 0;

int wordLength = 0;

int shortestWordLength = 100000000000;

int shortestWordIndex = 0;

int counter = 0;

int C = 2;

cout << "\n The length of all the words in the sentece : " << endl;

cout << "\n 1) ";

for (int x = 0; x <= l; x++)

{

if (str[x] == ' ' || str[x] == '\0')

{

cout << " - " << wordLength << ".\n " << C << ") ";

if (shortestWordLength > wordLength)

{

shortestWordLength = wordLength;

shortestWordIndex = counter - wordLength;

}

wordLength = 0;

counter++;

C++;

}

else

{

cout << str[x];

wordLength++;

counter++;

}

if (x == l)

cout << "\b\b\b\b " << endl;

}

cout << " The Length Of The Shortest Word : " << shortestWordLength << "." << endl;

cout << " The Index Of The Shortest Word : " << shortestWordIndex << "." << endl;

cout << "\n The First Shortest Word : ";

for (int y = shortestWordIndex; y < shortestWordIndex + shortestWordLength; y++)

{

cout << str[y];

}

cout << endl;

}

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

void line()

{

cout << "\n========================================================================================================================" << endl;

}

void cn()

{

int space = 0;

cout << "\n TO GO ANOTHER TASK, PRESS 1 . . . ";

cin >> space;

if (space == 1)

system("cls");

}

void main()

{

void (\*line\_ptr)() = line;

void (\*cn\_ptr)() = cn;

const int mysize = 100;

// Task 1

line\_ptr();

cout << "\n TASK 1" << endl;

cout << "\n Find the number of vowels in a string." << endl;

cout << " --------------------------------------" << endl;

char\* str1 = new char[mysize]{};

cout << "\n Enter a string : ";

cin.getline(str1, mysize);

int numberOfVowels = vowelCounter(str1 );

if (numberOfVowels != 0)

cout << "\n Number of vowels in this string is " << numberOfVowels << "." << endl;

else

cout << "\n There is no vowel in this string." << endl;

line\_ptr();

cn\_ptr();

// Task 2

line\_ptr();

cout << "\n TASK 2" << endl;

cout << "\n Find the number of consonants in a string." << endl;

cout << " ------------------------------------------" << endl;

cin.clear();

cin.ignore();

char\* str2 = new char[mysize]{};

cout << "\n Enter a string : ";

cin.getline(str2, mysize);

int numberOfConsonants = consonantCounter(str2);

if (numberOfConsonants != 0)

cout << "\n Number of consonants in this string is " << numberOfConsonants << "." << endl;

else

cout << "\n There is no consonant in this string." << endl;

line\_ptr();

cn\_ptr();

// Task 3

line\_ptr();

cout << "\n TASK 3" << endl;

cout << "\n Make the string PigLatin." << endl;

cout << " -------------------------" << endl;

cin.clear();

cin.ignore();

char\* str3 = new char[mysize]{};

cout << "\n Enter a string : ";

cin >> str3;

auto pigLatinChar = pigLatin(str3);

cout << "\n The Pig Latin form of the string is \'" << pigLatinChar << "\'." << endl;

line\_ptr();

cn\_ptr();

// Task 4

line\_ptr();

cout << "\n TASK 4" << endl;

cout << "\n Replace the spaces with the given symbol." << endl;

cout << " -----------------------------------------" << endl;

cin.clear();

cin.ignore();

char\* str4 = new char[mysize] {};

cout << "\n Enter a string : ";

cin.getline(str4,mysize);

char symbol = ' ';

cout << "\n Enter a symbol : ";

cin >> symbol;

auto newChar = replaceSpaces(str4, symbol);

cout << "\n String after spaces replaced with " << symbol << " : " << newChar << "." << endl;

line\_ptr();

cn\_ptr();

// Task 5

line\_ptr();

cout << "\n TASK 5" << endl;

cout << "\n Find whether one string is a suffix of other one." << endl;

cout << " -------------------------------------------------" << endl;

char\* suffix = new char[mysize] {};

cout << "\n Enter a suffix : ";

cin >> suffix;

cin.clear();

cin.ignore();

char\* str5 = new char[mysize] {};

cout << "\n Enter a string : ";

cin.getline(str5, mysize);

int result = isSuffix(suffix, str5);

if (result == 1)

cout << "\n \'" << suffix << "\' is a suffix of the word \'" << str5 << "\'." << endl;

else

cout << "\n \'" << suffix << "\' is not a suffix." << endl;

line\_ptr();

cn\_ptr();

// Task 6

line\_ptr();

cout << "\n TASK 6" << endl;

cout << "\n Find whether one string is a prefix of other one." << endl;

cout << " -------------------------------------------------" << endl;

char\* prefix = new char[mysize] {};

cout << "\n Enter a prefix : ";

cin >> prefix;

cin.clear();

cin.ignore();

char\* str6 = new char[mysize] {};

cout << "\n Enter a string : ";

cin.getline(str6, mysize);

int result2 = isPrefix(prefix, str6);

if (result2 == 1)

cout << "\n \'" << prefix << "\' is a prefix of the word \'" << str6 << "\'." << endl;

else

cout << "\n \'" << prefix << "\' is not a prefix." << endl;

line\_ptr();

cn\_ptr();

// Task 7

line\_ptr();

cout << "\n TASK 7" << endl;

cout << "\n Find whether a string is a palindrome." << endl;

cout << " --------------------------------------" << endl;

cin.clear();

cin.ignore();

char\* str7 = new char[mysize] {};

cout << "\n Enter a string : ";

cin.getline(str7, mysize);

int result3 = isPalindrome(str7);

if (result3 == 1)

cout << "\n The string \'" << str7 << "\' is a palindrome." << endl;

else

cout << "\n The string \'" << str7 << "\' is not a palindrome." << endl;

line\_ptr();

cn\_ptr();

// Task 8

line\_ptr();

cout << "\n TASK 8" << endl;

cout << "\n Find the number of numerals, letters, punctuation marks in a string." << endl;

cout << " --------------------------------------------------------------------" << endl;

cin.clear();

cin.ignore();

char\* str8 = new char[mysize] {};

cout << "\n Enter a string : ";

cin.getline(str8, mysize);

int numbersQuantity = numberCounter(str8);

int lettersQuantity = letterCounter(str8);

int punctuationQuantity = punctuationCounter(str8);

if (numbersQuantity > 1)

cout << "\n There are " << numbersQuantity << " numbers in the string." << endl;

else if (numbersQuantity == 1)

cout << "\n There is only 1 number in the string." << endl;

else

cout << "\n There is no number in the string." << endl;

if (lettersQuantity > 1)

cout << " There are " << lettersQuantity << " letters in the string." << endl;

else if (lettersQuantity == 1)

cout << " There is only 1 letter in the string." << endl;

else

cout << " There is no letter in the string." << endl;

if (punctuationQuantity > 1)

cout << " There are " << punctuationQuantity << " punctuation marks in the string." << endl;

else if (punctuationQuantity == 1)

cout << " There is only 1 punctuation mark in the string." << endl;

else

cout << " There is no punctuation mark in the string." << endl;

line\_ptr();

cn\_ptr();

// Task 9

line\_ptr;

cout << "\n TASK 9" << endl;

cout << "\n Find the number of \"the\"s in a string." << endl;

cout << " --------------------------------------" << endl;

cin.clear();

cin.ignore();

char\* str9 = new char[mysize] {};

cout << "\n Enter a string : ";

cin.getline(str9, mysize);

int countOfTHE = theCounter(str9);

if (countOfTHE != 0)

cout << "\n Number of -the- in the string is " << countOfTHE << "." << endl;

else

cout << "\n The is no word \"the\" in a string." << endl;

line\_ptr();

cn\_ptr();

// Task 10

line\_ptr();

cout << "\n TASK 10" << endl;

cout << "\n Delete the numbers and punctuation marks in a string." << endl;

cout << " -----------------------------------------------------" << endl;

cin.clear();

cin.ignore();

char\* str10 = new char[mysize] {};

cout << "\n Enter a string : ";

cin.getline(str10, mysize);

char\* newArray = deleteNumberAndPunct(str10);

cout << "\n String after numbers and punctuation marks deleted : \'" << newArray << "\'." << endl;

line\_ptr();

cn\_ptr();

// Task 11

line\_ptr();

cout << "\n TASK 11" << endl;

cout << "\n Find the most used letter in a string." << endl;

cout << " --------------------------------------" << endl;

cin.clear();

cin.ignore();

char\* str11 = new char[mysize] {};

cout << "\n Enter a sentence : ";

cin.getline(str11,mysize);

char mostUsedLetterV = mostUsedLetter(str11);

cout << "\n The most used letter in the sentence is : " << mostUsedLetterV << endl;

line\_ptr();

cn\_ptr();

// Task 12

line\_ptr();

cout << "\n TASK 12" << endl;

cout << "\n Find the longest word in a string." << endl;

cout << " ----------------------------------" << endl;

cin.clear();

cin.ignore();

char\* str12 = new char[mysize] {};

cout << "\n Enter a sentence : ";

cin.getline(str12, mysize);

longestWord(str12);

line\_ptr();

cn\_ptr();

// Task 13

line\_ptr();

cout << "\n TASK 13" << endl;

cout << "\n Find the shortest word in a string." << endl;

cout << " -----------------------------------" << endl;

cin.clear();

cin.ignore();

char\* str13 = new char[mysize] {};

cout << "\n Enter a sentence : ";

cin.getline(str13, mysize);

shortestWord(str13);

line\_ptr();

cn\_ptr();

line\_ptr();

cout << "\n Tasks ended." << endl;

line\_ptr();

}