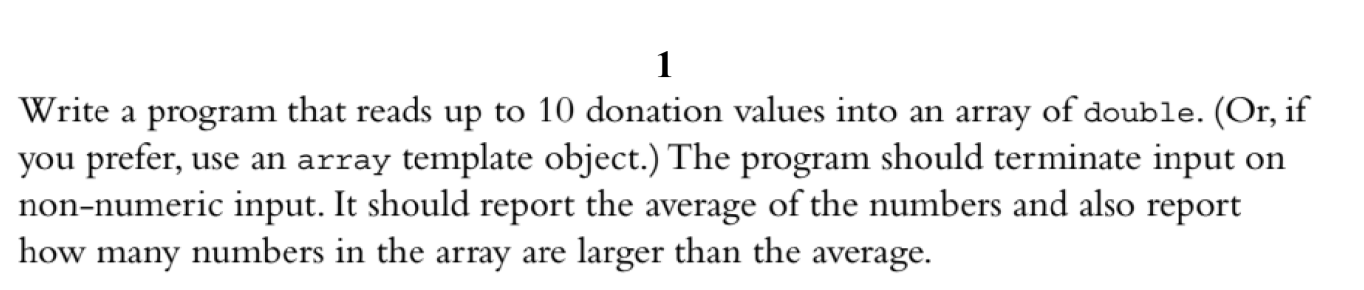
Oleynik Vladislav

Exercise 9\_10



Code:

#include <iostream>

#include <list>

using namespace std;

list<double> list\_of\_numbers;

double average\_value = 0;

void logValue (double value) {

cout << "================================" << endl;

cout << "Average value is " << average\_value << endl;

int quantity\_greater\_avr\_value = 0;

for(double x : list\_of\_numbers) {

if (x > average\_value) quantity\_greater\_avr\_value++;

}

cout << quantity\_greater\_avr\_value << " value greater then average\_value" << endl;

cout << "================================" << endl;

cout << endl;

}

double calculate\_average\_value () {

double avr\_val = 0;

int i = 0;

for (double x : list\_of\_numbers) {

i++;

avr\_val += x;

}

average\_value = avr\_val / i;

}

void enterValuesInToArray() {

for(int i=0; i <= 10; i++) {

cout << "Enter number with " << i << " index:" << endl;

double value\_to\_push;

cin >> value\_to\_push;

list\_of\_numbers.push\_back(value\_to\_push);

calculate\_average\_value();

logValue(value\_to\_push);

}

}

int main()

{

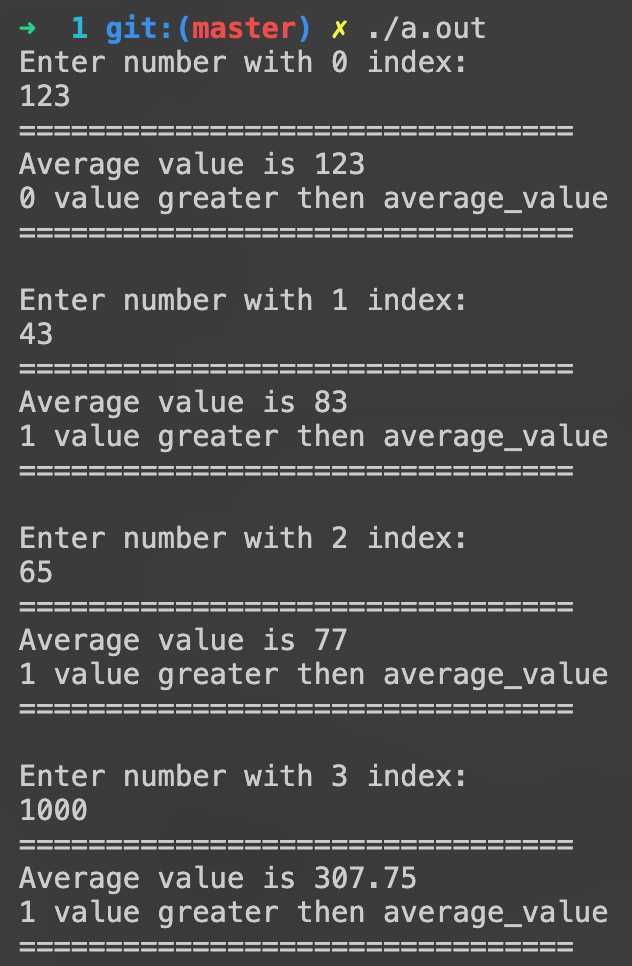
enterValuesInToArray();

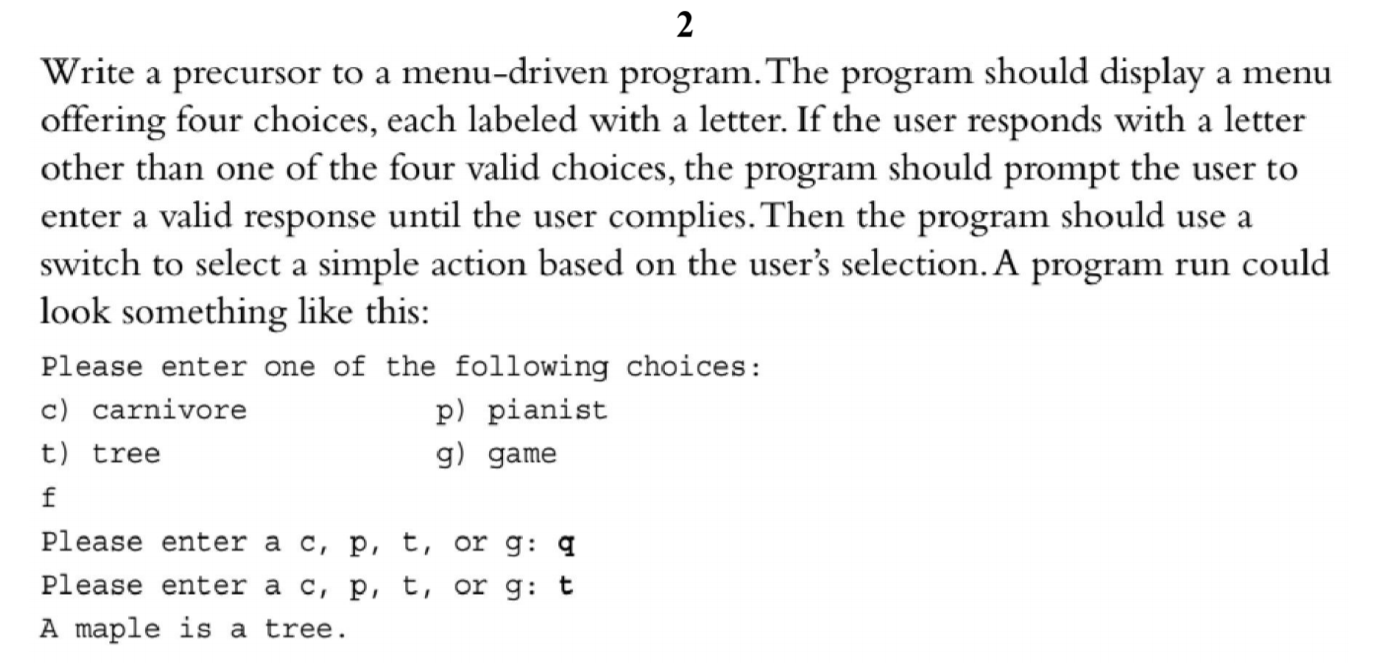
cout << average\_value << endl;

return 0;

}

========Output========





Code:

#include <iostream>

#include <list>

using namespace std;

struct option

{

string option;

string label;

string output;

} option1, option2, option3, option4;

void display\_option(string option, string label)

{

cout << option << ") " << label << endl;

}

void display\_options()

{

display\_option(option1.option, option1.label);

display\_option(option2.option, option2.label);

display\_option(option3.option, option3.label);

cout << endl;

}

void choose\_options()

{

string choosen\_option;

while (true)

{

cout << "Please, choose a color from menu below:" << endl;

display\_options();

cin >> choosen\_option;

if (choosen\_option == option1.option)

{

cout << option1.output << endl;

return;

}

else if (choosen\_option == option2.option)

{

cout << option2.output << endl;

return;

}

else if (choosen\_option == option3.option)

{

cout << option3.output << endl;

return;

}

else

{

cout << "no, please, enter valid variant " << endl;

}

}

}

int main()

{

option1.option = "a";

option1.label = "red";

option1.output = "wooow, you choose red color";

option2.option = "b";

option2.label = "green";

option2.output = "hm, green is good color";

option3.option = "c";

option3.label = "blue";

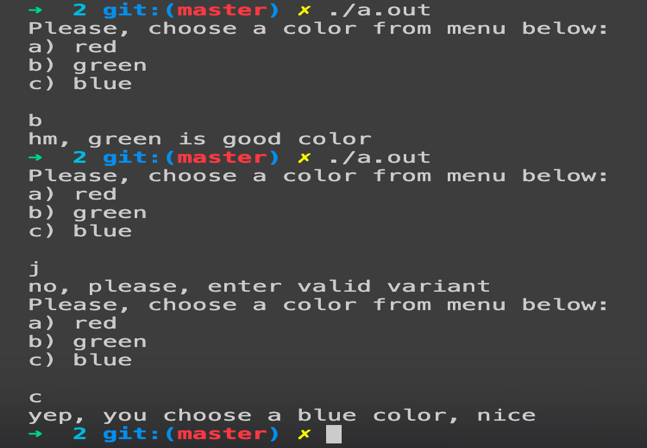
option3.output = "yep, you choose a blue color, nice";

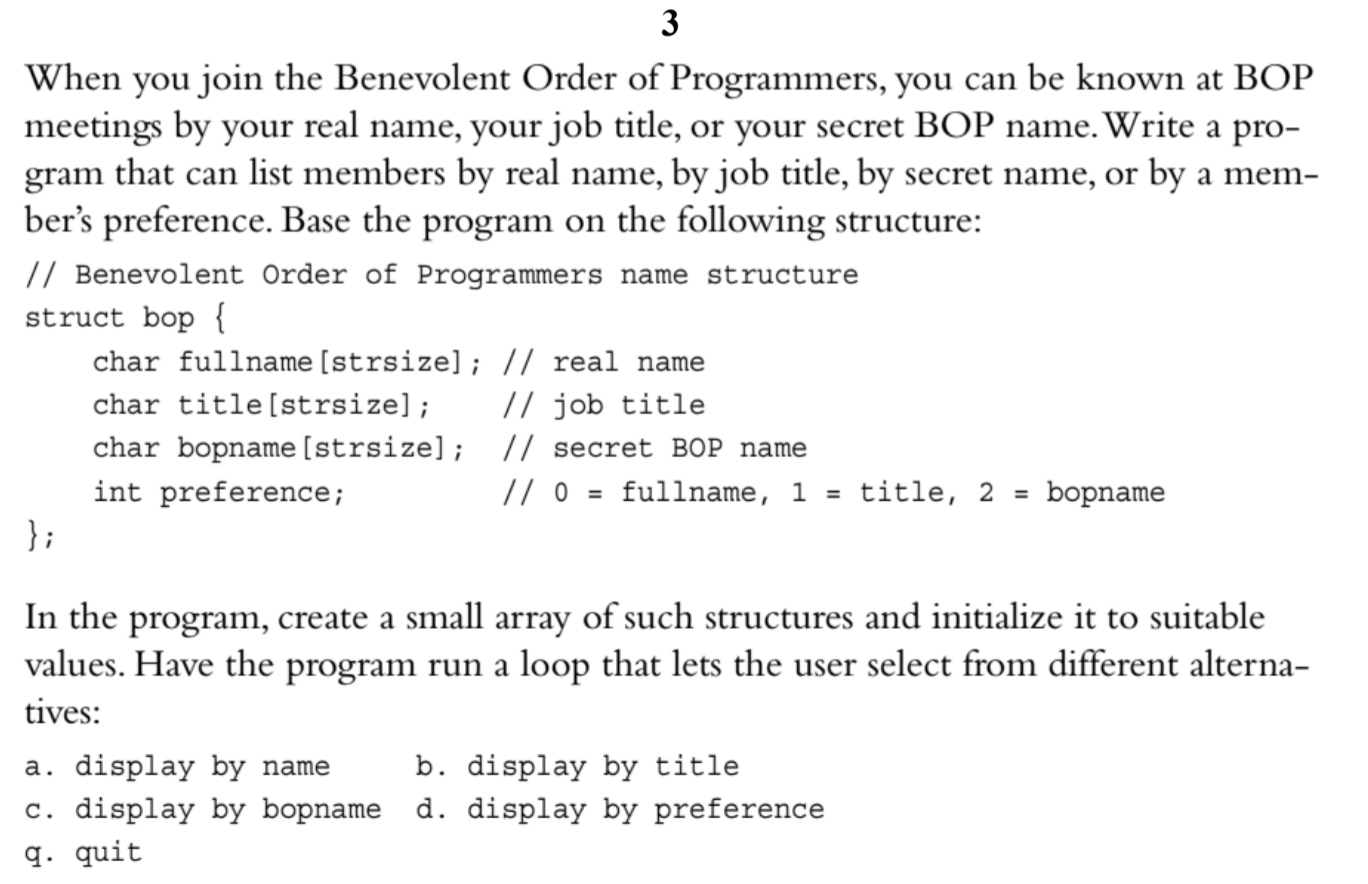
choose\_options();

return 0;

}

Output:





Code:

#include <iostream>

#include <list>

using namespace std;

struct bop

{

string fullname;

string title;

string bopname;

int preference;

};

bop bops[5] = {

{"Riley Strickland", "Hihihi, how are you", "realy", 0},

{"Donavan Gray", "what are you doing? ", "donaldtrump", 2},

{"Kane King", "I'm king!!!", "mirinda", 1},

{"Brielle Owen", "loool, what's up dude", "own", 0},

{"Riley Hampton", "I'm relay, hahah", "relay", 2},

};

int current\_choose;

void output\_variant\_for\_display()

{

cout << "0. display by name" << endl;

cout << "1. display by title" << endl;

cout << "2. display by bopname" << endl;

cout << "3. display by preference" << endl;

cout << "5. quit" << endl;

}

void output\_value(int index, int preference)

{

if (preference == 0)

{

cout << bops[index].bopname;

}

else if (preference == 1)

{

cout << bops[index].title;

}

else if (preference == 2)

{

cout << bops[index].bopname;

}

else if (preference == 3)

{

cout << bops[index].bopname;

}

cout << endl;

}

void output\_values(int preference)

{

cout << "========" << endl;

for (int i = 0; i < sizeof(bops) / sizeof(bops[0]); i++)

{

output\_value(i, preference);

}

cout << "========" << endl;

}

bool input\_variant()

{

cout << "Enter your choice: " << endl;

cin >> current\_choose;

if (current\_choose == 5)

{

return true;

}

output\_values(current\_choose);

return false;

}

int main()

{

while (true)

{

output\_variant\_for\_display();

bool result = input\_variant() ? false : true;

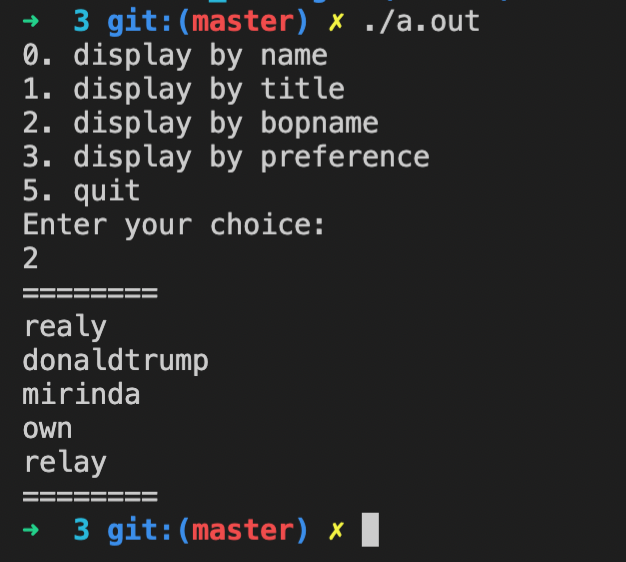
if (result)

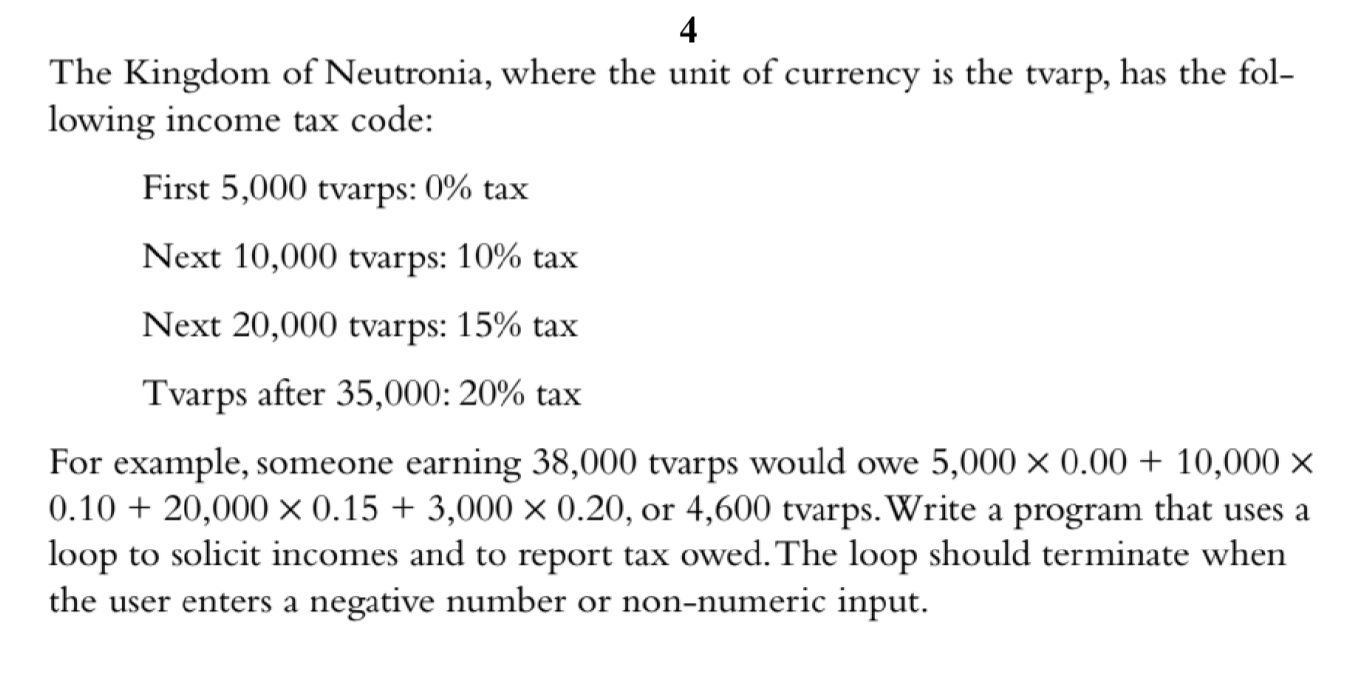
break;

}

}

Output:





#include <iostream>

#include <list>

using namespace std;

double value\_to\_calculate;

struct taxes

{

double first;

double next1;

double next2;

double last;

};

taxes values\_of\_taxes;

void input\_value()

{

cout << "Please input value to see calculation results" << endl;

cin >> value\_to\_calculate;

}

void sort\_by\_steps()

{

value\_to\_calculate = value\_to\_calculate - 5000;

values\_of\_taxes.first = value\_to\_calculate;

value\_to\_calculate = value\_to\_calculate - (10000 - 5000);

values\_of\_taxes.next1 = value\_to\_calculate;

value\_to\_calculate = value\_to\_calculate - (20000 - 10000);

values\_of\_taxes.next2 = value\_to\_calculate;

value\_to\_calculate = value\_to\_calculate - (35000 - 2000);

values\_of\_taxes.last = value\_to\_calculate;

}

void calculate\_and\_output\_relust()

{

double calculated\_value = 0;

sort\_by\_steps();

if (values\_of\_taxes.first > 0)

{

calculated\_value += values\_of\_taxes.first;

}

if (values\_of\_taxes.next1 > 0)

{

calculated\_value += values\_of\_taxes.next1;

}

if (values\_of\_taxes.next2 > 0)

{

calculated\_value += values\_of\_taxes.next2;

}

if (values\_of\_taxes.last > 0)

{

calculated\_value += values\_of\_taxes.last;

}

cout << "After taxes clear value is " << calculated\_value << endl;

}

int main()

{

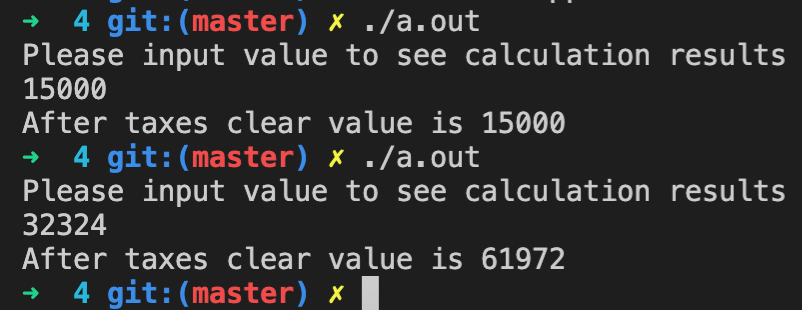
input\_value();

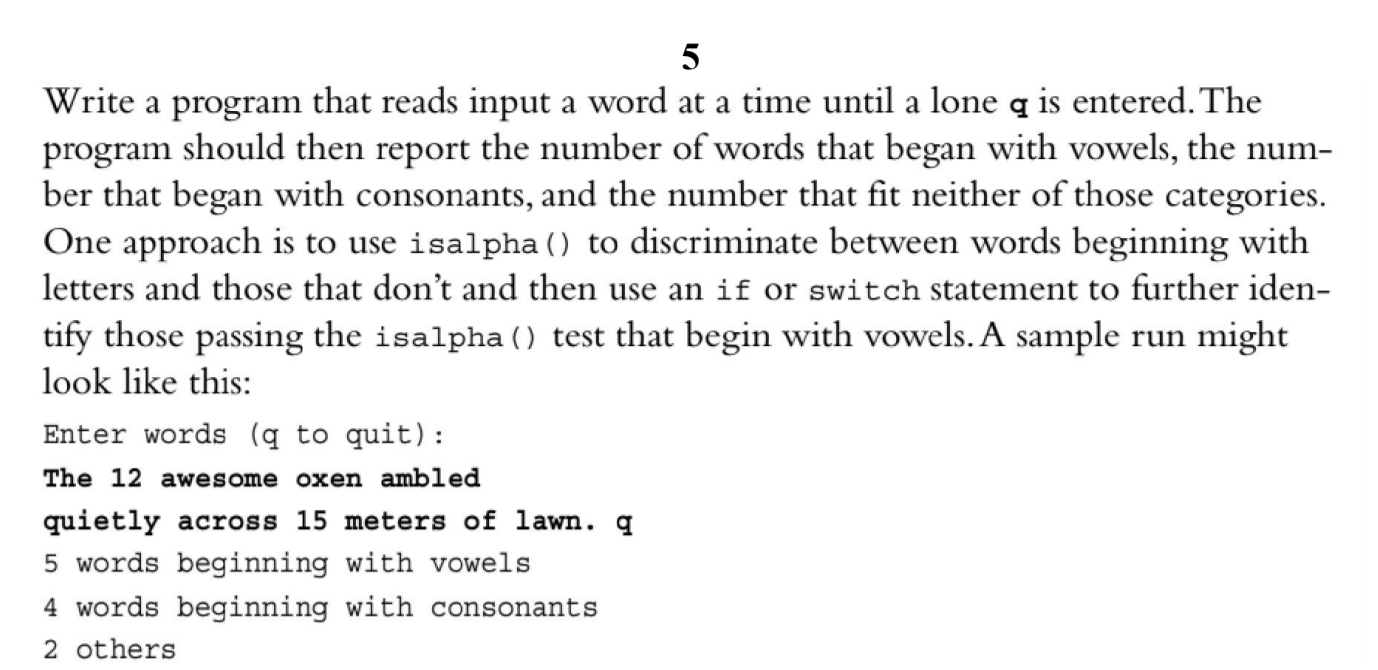
calculate\_and\_output\_relust();

return 0;

}

Output:





#include <iostream>

#include <string>

#include <cctype>

using namespace std;

int main()

{

int others = 0,

vowels = 0,

consonants = 0;

string word;

cout << "Enter words (q to quit)\n";

// read a word at a time till a single 'q'

while ( cin >> word && word != "q" )

{

// consider only the first letter of the word

char c = word[0];

if ( isalpha(c) )

{

if ( c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u' ||

c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U' )

{

++vowels;

}

else

{

++consonants;

}

}

else

{

++others;

}

}

cout << vowels << " words beginning with vowels\n";

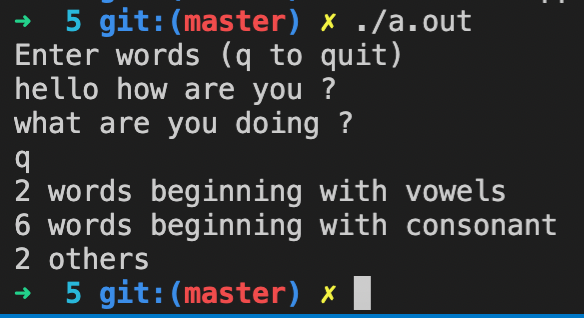
cout << consonants << " words beginning with consonant\n";

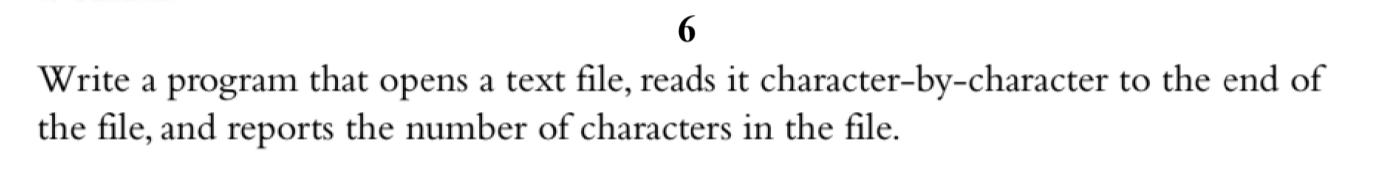
cout << others << " others\n";

return 0;

}

Output:





#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int main()

{

// Create a text string, which is used to output the text file

string myText;

// Read from the text file

ifstream MyReadFile("text.txt");

// Use a while loop together with the getline() function to read the file line by line

while (getline(MyReadFile, myText))

{

// Output the text from the file

cout << myText << endl;

cout << "Size: " << myText.size() << endl;

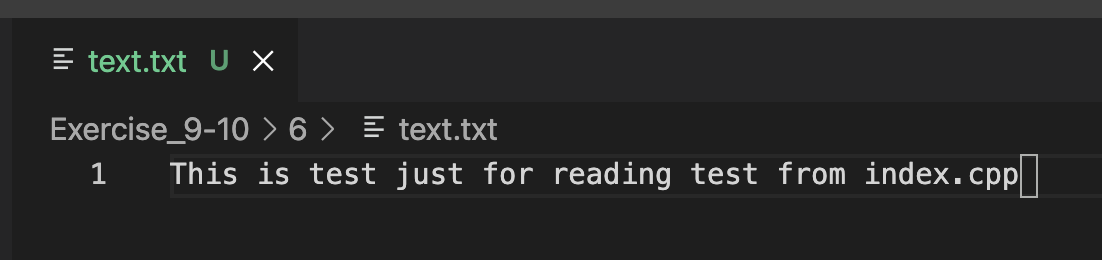
}

// Close the file

MyReadFile.close();

}

.txt file:



Output:

