Graphical user interface, text, chat or text message

Description automatically generated

Main.cpp

// Kenry Yu, Olenka Bilinska, Rushil Prajapati

// Demo @ 5:10

// Week 8 Lab 2

#include "person.h"

#include <iostream>

using namespace std;

const int SIZE = 100;

void bsort(person \*[], int);

int main() {

person \*persPtr[SIZE];

int n = 0;

char choice = '\0';

do {

persPtr[n] = new person; // make new object

persPtr[n]->setName(); // set person's names

n++; // count new person

cout << "Enter another (y/n)? "; // enter another

cin >> choice;

} while (choice != 'N' && choice != 'n');

cout << "\nUnsorted list:" << endl;

for (int i = 0; i < n; i++)

persPtr[i]->printName();

cout << "\nSorted list:" << endl;

bsort(persPtr, n);

for (int i = 0; i < n; i++)

persPtr[i]->printName();

return 0;

}

void bsort(person \*names[], int size) {

for (int i = 0; i < size - 1; i++)

if (int(names[i]->getName()[0]) > int(names[i + 1]->getName()[0]))

for (int j = i + 1; j > 0; j--) {

if (int(names[j]->getName()[0]) < int(names[j - 1]->getName()[0])) {

person \*temp = names[j - 1];

names[j - 1] = names[j];

names[j] = temp;

} else {

break;

}

}

}

Person.h

#include <iostream>

#include <string> //for string class

using namespace std;

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

class person // class of persons

{

protected:

string name; // person's name

public:

void setName() // set the name

{

cout << "Enter name: ";

cin >> name;

}

void printName() // display the name

{

cout << name << endl;

}

string getName() // return the name

{

return name;

}

};

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