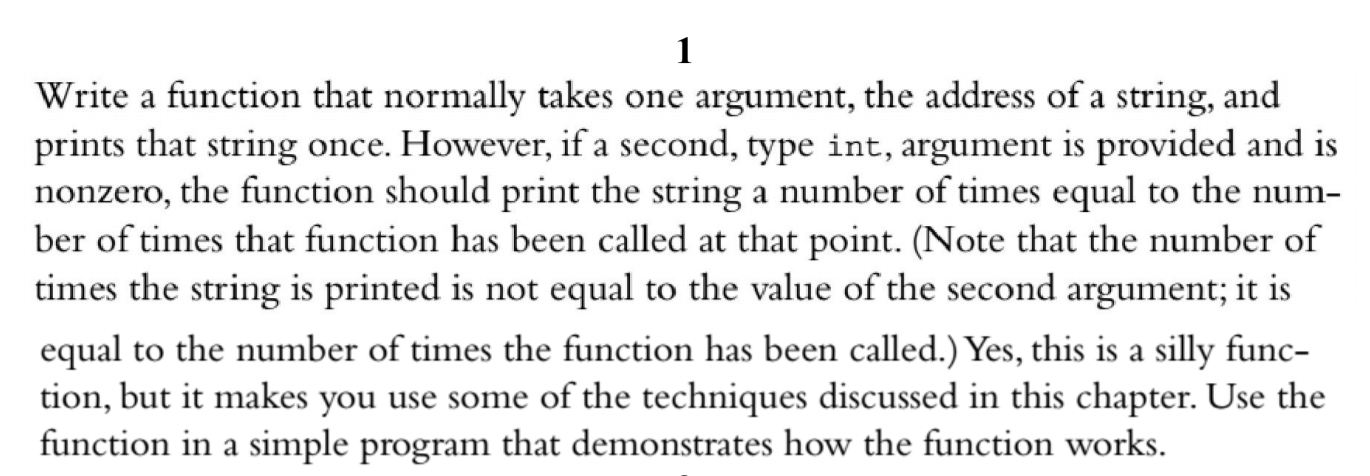
Oleynik Vladislav

Exercise 13



#include <iostream>

using namespace std;

int scores[10];

int avarage\_score = 0;

void print(string string, int quantity = 1)

{

{

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

{

cout << string << endl;

}

}

}

int main()

{

print("HELLO", 1);

print("HELLO", 3);

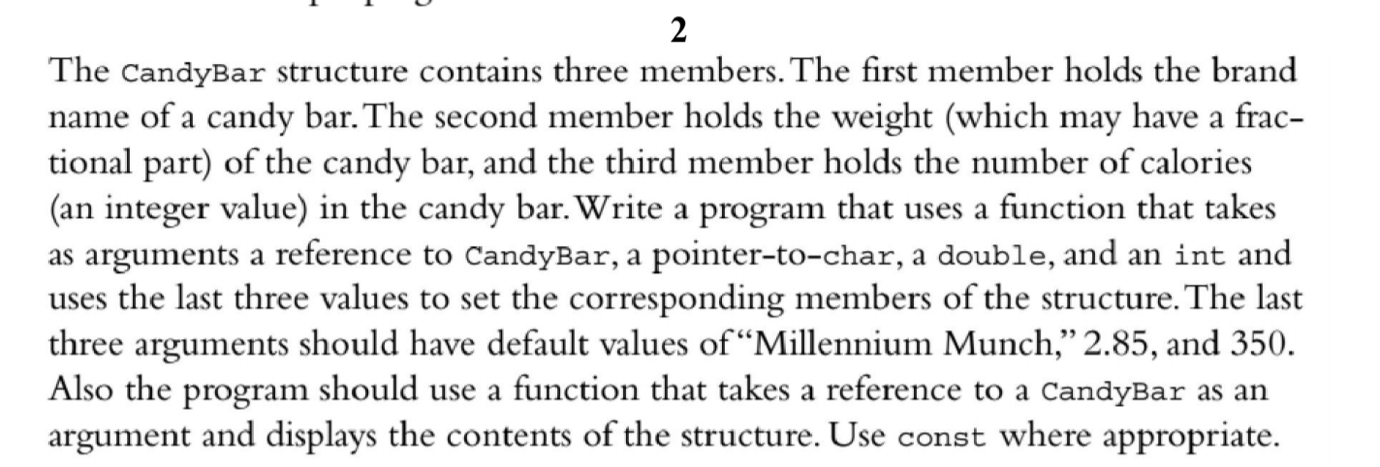
print("HELLO", 6);

return 0;

}

Output:





#include <iostream>

using namespace std;

struct CandyBar

{

char \*brand;

double weight;

int calories;

};

void setValues(CandyBar &candyBar, char \*brand = "Apple", double weight = 222, int calories = 13)

{

candyBar.brand = brand;

candyBar.weight = weight;

candyBar.calories = calories;

}

void printCandyBar(const CandyBar &candyBar)

{

cout << "Brand: " << candyBar.brand << endl;

cout << "Weight: " << candyBar.weight << endl;

cout << "Calories: " << candyBar.calories << endl;

}

int main()

{

CandyBar candyBar;

cout << "=== Set and show default values ===" << endl;

setValues(candyBar);

printCandyBar(candyBar);

cout << "=== Set and show non-default values ===" << endl;

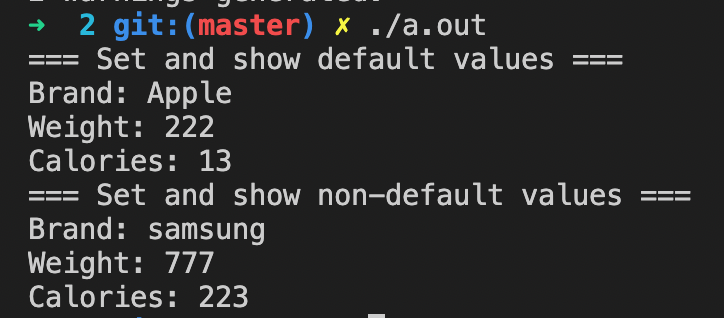
setValues(candyBar, "samsung", 777, 223);

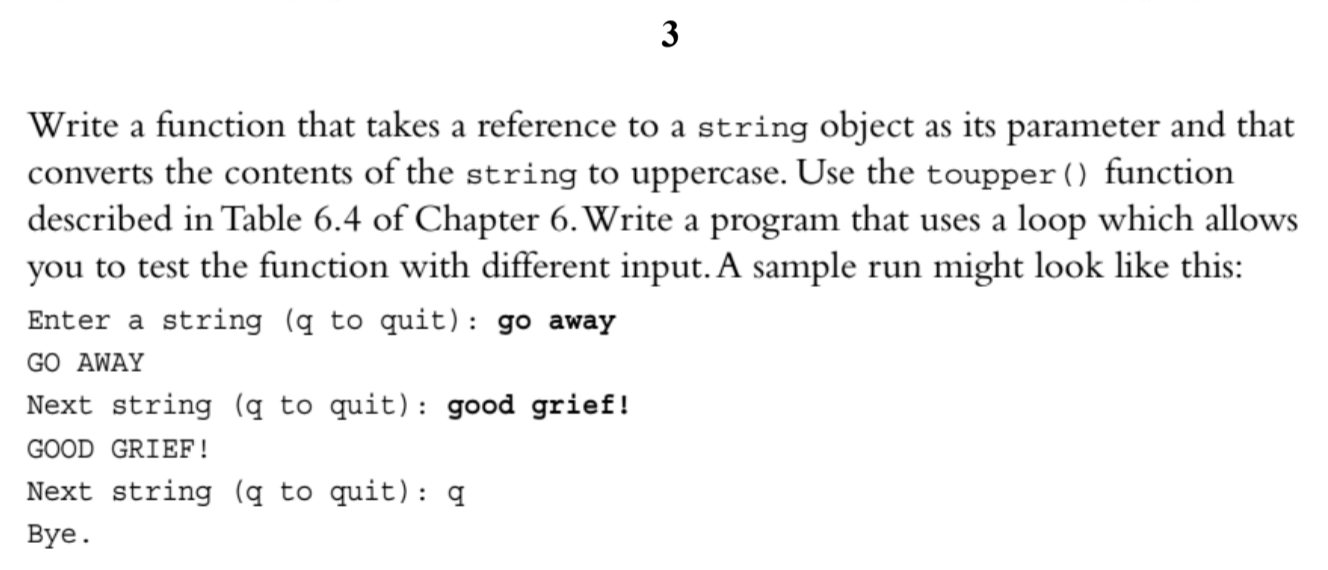
printCandyBar(candyBar);

return 0;

}

Output:





#include <iostream>

#include <string>

#include <cctype>

using namespace std;

void toUppercase(string &string1)

{

for (unsigned i = 0; i < string1.size(); i++)

{

string1[i] = toupper(string1[i]);

}

}

int main()

{

string string1;

cout << "Enter a string (q to quit): ";

while (true)

{

cin >> string1;

if (string1 == "q")

{

break;

}

toUppercase(string1);

cout << string1 << endl;

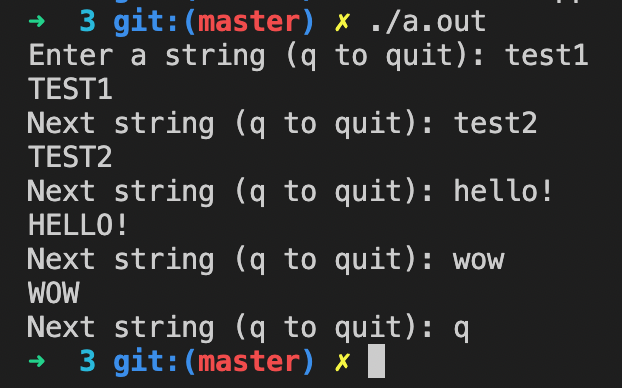
cout << "Next string (q to quit): ";

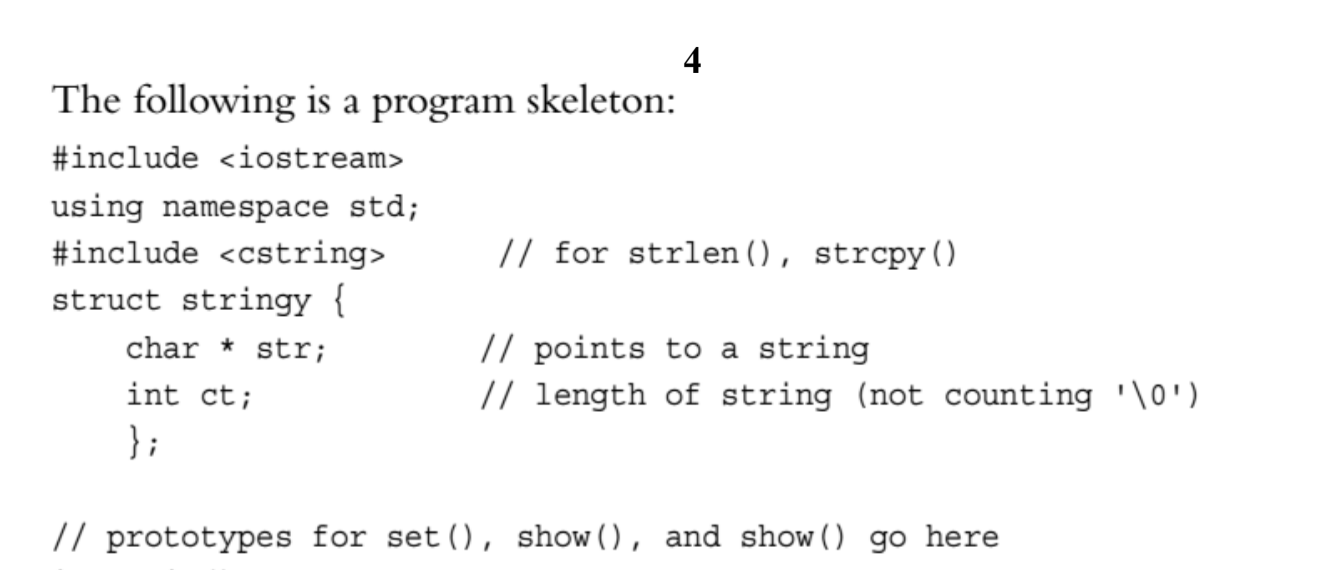
}

return 0;

}

Output:





#include <iostream>

#include <cstring>

using namespace std;

struct stringy

{

char \*str;

int ct;

};

void set(stringy &, const char \*);

void show(const char \*, int n = 1);

void show(const stringy &, int n = 1);

int main()

{

stringy beany;

char testing[] = "Reality isn't what it used to be.";

set(beany, testing);

show(beany);

show(beany, 2);

testing[0] = 'D';

testing[1] = 'u';

show(testing);

show(testing, 3);

show("Done!");

return 0;

}

void set(stringy &aStringy, const char \*aString)

{

int stringLength = sizeof(aString) / sizeof(char);

aStringy.str = new char[stringLength];

aStringy.ct = stringLength;

strcpy(aStringy.str, aString);

}

void show(const char \*aString, int n)

{

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

{

cout << aString << endl;

}

}

void show(const stringy &aStringy, int n)

{

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

{

cout << aStringy.str << endl;

}

}

Output:

