Início	quarta, 25 de maio de 2022 às 08:36
Estado	Prova submetida
Data de submissão:	segunda, 30 de maio de 2022 às 15:45
Tempo gasto	5 dias 7 horas
Nota	**100** do máximo 100

Pergunta 1

Correta Pontuou 20 de 20

Write a C++ function `unsigned roman_to_arab(const string& roman)`, that, given a valid roman numeral in the standard form (as a string), returns its corresponding decimal value (as an `unsigned` integer).

Use the following dictionary, represented in the appropriate STL container, to map each individual roman symbol to its decimal value:

```
map<char, unsigned> map_roman = {
    {'I', 1},
    {'V', 5},
    {'X', 10},
    {'L', 50},
    {'C', 100},
    {'D', 500},
    {'M', 1000}
};
```

Por exemplo:

Teste	Resultado

`string r = "XV";`
`cout << roman_to_arab(r) << '\n';`
	15

`string r = "LXXXIV";`
`cout << roman_to_arab(r) << '\n';`
	84

`string r = "CMLXIV";`
`cout << roman_to_arab(r) << '\n';`
	964

`string r = "MMMCMXCIX";`
`cout << roman_to_arab(r) << '\n';`
	3999

`string r = "MMMDCCCLXXXVIII";`
`cout << roman_to_arab(r) << '\n';`
	3888

Resposta: (regime de penalização: 0, 0, 0, 0, 10, 20, 30, ... %)

Limpar resposta

```
#include <iostream>
#include <map>

using namespace std;

unsigned roman_to_arab(const string& roman){
    unsigned result = 0;
    map<char, unsigned> map_roman = {
        {'I', 1},
        {'V', 5},
        {'X', 10},
        {'L', 50},
        {'C', 100},
        {'D', 500},
        {'M', 1000}
    };

    unsigned current = 1000;
    for(char c: roman){
        unsigned value = map_roman[c];
        if(value <= current){
            result += value;
```

	Teste	Esperado	Recebido	

✓	`string r = "XV";`
	`cout << roman_to_arab(r) << '\n';`
		15	15	✓

✓	`string r = "LXXXIV";`
	`cout << roman_to_arab(r) << '\n';`
		84	84	✓

✓	`string r = "CMLXIV";`
	`cout << roman_to_arab(r) << '\n';`
		964	964	✓

	Teste	Esperado	Recebido	
✓	`string r = "MMMCMXCIX";` `cout << roman_to_arab(r) << '\n';`	3999	3999	✓
✓	`string r = "MMMDCCCLXXXVIII";` `cout << roman_to_arab(r) << '\n';`	3888	3888	✓

Passou em todos os testes! ✓

Solução do autor da pergunta (C):

```
#include <iostream>
#include <map>
#include <regex>  // this was not required, but I love regex

using namespace std;

//! Converts roman numbers to arabic numbers
unsigned roman_to_arab(const string& s) {
  // this was not required, but I like regex
  if (!regex_match(s, regex("^M{0,3}(CM|CD|D?C{0,3})(XC|XL|L?X{0,3})(IX|IV|V?I{0,3})$")))
    cout << ">> THIS IS A WRONG ROMAN! ";

  // the dictionary, local but created only once
  static map<char, unsigned> map_roman = {
    {'I', 1},
    {'V', 5},
    {'X', 10},
    {'L', 50},
    {'C', 100},
    {'D', 500},
    {'M', 1000}
  };
```

Correta

Nota desta submissão: 20/20

Pergunta 2

Correta Pontuou 20 de 20

Write a C++ function `void count_words(const string& str, map<string, size_t>& count)`, that receives as parameters a `string` and a `map`, and fills the `map` with the number of occurrences of each word in the `string`. Consider the words case insensitive and the keys in lowercase.

Use the function `show_map`, given in preload, to show the contents of the `map`.

Por exemplo:

Teste	Resultado

`string s = "If you want to buy  buy  if you don't want to buy  bye bye";`
`map<string, size_t> count;`
`count_words(s, count);`
`show_map(count);`
	`[ buy:3 bye:2 don't:1 if:2 to:2 want:2 you:2 ]`

`string s = "You can fool some of the people all of the time and all of the people some of the time  but you cannot fool all of the people all of the time";`
`map<string, size_t> count;`
`count_words(s, count);`
`show_map(count);`
	`[ all:4 and:1 but:1 can:1 cannot:1 fool:2 of:6 people:3 some:2 the:6 time:3 you:2 ]`

`string s = "A tutor who tooted the flute tried to tutor two young tooters to toot   Said the two to the tutor is it tougher to toot or to tutor two young tooters to toot";`
`map<string, size_t> count;`
`count_words(s, count);`
`show_map(count);`
	`[ a:1 flute:1 is:1 it:1 or:1 said:1 the:3 to:6 toot:3 tooted:1 tooters:2 tougher:1 tried:1 tutor:4 two:3 who:1 young:2 ]`

`string s = "You do not need to turn on a night light on a clear night like tonight because in a clear night there is always a light light and tonight is a clear night";`
`map<string, size_t> count;`
`count_words(s, count);`
`show_map(count);`
	`[ a:5 always:1 and:1 because:1 clear:3 do:1 in:1 is:2 light:3 like:1 need:1 night:4 not:1 on:2 there:1 to:1 tonight:2 turn:1 you:1 ]`

`string s = "If one doctor doctors another doctor  Does the doctor who doctors the doctor  Doctor the way the doctor he is doctoring doctor Or does he doctor the doctor  The way the doctor who doctors doctor";`
`map<string, size_t> count;`
`count_words(s, count);`
`show_map(count);`
	`[ another:1 doctor:11 doctoring:1 doctors:3 does:2 he:2 if:1 is:1 one:1 or:1 the:7 way:2 who:2 ]`

Resposta: (regime de penalização: 0, 0, 0, 0, 10, 20, 30, ... %)

Limpar resposta

```cpp
#include <iostream>
#include <string>
#include <sstream>
#include <iomanip>
#include <map>

using namespace std;

//! print map in one line
void show_map(const map<string, size_t>& count) {
  cout << "[ ";
  for (const auto& e : count)   {
    cout << e.first << ":" << e.second << ' ';
  }
  cout << "]\n";
}

void count_words(const string& str, map<string, size_t>& count){
    istringstream stream(str);
    string it;
    while(stream >> it){
        for(char& c: it) c = tolower(c);
```

	Teste	Esperado	Recebido	

✓	`string s = "If you want to buy  buy  if you don't want to buy  bye bye";`
	`map<string, size_t> count;`
	`count_words(s, count);`
	`show_map(count);`
		`[ buy:3 bye:2 don't:1 if:2 to:2 want:2 you:2 ]`	`[ buy:3 bye:2 don't:1 if:2 to:2 want:2 you:2 ]`	✓

✓	`string s = "You can fool some of the people all of the time and all of the people some of the time  but you cannot fool all of the people all of the time";`
	`map<string, size_t> count;`
	`count_words(s, count);`
	`show_map(count);`
		`[ all:4 and:1 but:1 can:1 cannot:1 fool:2 of:6 people:3 some:2 the:6 time:3 you:2 ]`	`[ all:4 and:1 but:1 can:1 cannot:1 fool:2 of:6 people:3 some:2 the:6 time:3 you:2 ]`	✓

✓	`string s = "A tutor who tooted the flute tried to tutor two young tooters to toot Said the two to the tutor is it tougher to toot or to tutor two young tooters to toot";`
	`map<string, size_t> count;`
	`count_words(s, count);`
	`show_map(count);`
		`[ a:1 flute:1 is:1 it:1 or:1 said:1 the:3 to:6 toot:3 tooted:1 tooters:2 tougher:1 tried:1 tutor:4 two:3 who:1 young:2 ]`	`[ a:1 flute:1 is:1 it:1 or:1 said:1 the:3 to:6 toot:3 tooted:1 tooters:2 tougher:1 tried:1 tutor:4 two:3 who:1 young:2 ]`	✓

✓	`string s = "You do not need to turn on a night light on a clear night like tonight because in a clear night there is always a light light and tonight is a clear night";`
	`map<string, size_t> count;`
	`count_words(s, count);`
	`show_map(count);`
		`[ a:5 always:1 and:1 because:1 clear:3 do:1 in:1 is:2 light:3 like:1 need:1 night:4 not:1 on:2 there:1 to:1 tonight:2 turn:1 you:1 ]`	`[ a:5 always:1 and:1 because:1 clear:3 do:1 in:1 is:2 light:3 like:1 need:1 night:4 not:1 on:2 there:1 to:1 tonight:2 turn:1 you:1 ]`	✓

✓	`string s = "If one doctor doctors another doctor  Does the doctor who doctors the doctor  Doctor the way the doctor he is doctoring doctor  Or does he doctor the doctor  The way the doctor who doctors doctor";`
	`map<string, size_t> count;`
	`count_words(s, count);`
	`show_map(count);`
		`[ another:1 doctor:11 doctoring:1 doctors:3 does:2 he:2 if:1 is:1 one:1 or:1 the:7 way:2 who:2 ]`	`[ another:1 doctor:11 doctoring:1 doctors:3 does:2 he:2 if:1 is:1 one:1 or:1 the:7 way:2 who:2 ]`	✓

Passou em todos os testes! ✓

Solução do autor da pergunta (C):

```cpp
#include <iostream>
#include <string>
#include <sstream>
#include <iomanip>
#include <map>

using namespace std;

//! print map in one line
void show_map(const map<string, size_t>& count) {
  cout << "[ ";
  for (const auto& e : count)   {
    cout << e.first << ":" << e.second << ' ';
  }
  cout << "]\n";
}

void count_words(const string& str, map<string, size_t>& count) {
  istringstream iss(str);
  string word;

  count.clear(); // guarantee that counts start at zero
```

Correta

Nota desta submissão: 20/20

Pergunta 3

Correta Pontuou 20 de 20

Consider a class named `MovingAverage` in header MovingAverage.h with the following interface:

```
class MovingAverage {
public:
  MovingAverage(std::size_t n);
  void update(double value);
  double get() const;
private:
  std::size_t n_;
  std::list<double> values_;
};
```

As the name indicates, the purpose of the class is to allow the computation of moving averages:

- The values calculated will be the average of (at most) `n_` values, where the initial value for `n_` is specified using the constructor.
- A call to `update(v)` sets the most recent value to consider for the moving average, adding `v` to the `values_` list. If there are already `n_` values stored, the least recent value must be discarded (removed from `values_`).
- A call to `get()` returns the value of the moving average considering the currently stored values.

Provide the implementation of the class in a file called `MovingAverage.cpp`.

Hint: consider the use of `pop_front()` and `push_back()` to manipulate the list.

Por exemplo:

Teste	Resultado

`MovingAverage ma(1);`
`const MovingAverage& r = ma;`
`ma.update(1.0);`
`cout << fixed << setprecision(2) << r.get() << '\n';`
	`1.00`

`MovingAverage ma(1);`
`double a[] { 1.2, 3.4, 5.0 };`
`for (double x : a) {`
`  ma.update(x);`
`  cout << fixed << setprecision(2) <<  ma.get() << ' ';`
`}`
`cout << fixed << setprecision(2) <<  ma.get() << '\n';`
	`1.20 3.40 5.00 5.00`

`MovingAverage ma(2);`
`double a[] { 1.2, 3.4, 5.0 };`
`for (double x : a) {`
`  ma.update(x);`
`  cout << fixed << setprecision(2) <<  ma.get() << ' ';`
`}`
`cout << fixed << setprecision(2) <<  ma.get() << '\n';`
	`1.20 2.30 4.20 4.20`

`MovingAverage ma(3);`
`double a[] { 1.2, 3.4, 5.0 };`
`for (double x : a) {`
`  ma.update(x);`
`  cout << fixed << setprecision(2) <<  ma.get() << ' ';`
`}`
`cout << fixed << setprecision(2) <<  ma.get() << '\n';`
	`1.20 2.30 3.20 3.20`

`MovingAverage ma(4);`
`double a[] { 0.1, 1.2, -1.2, 3.4, 5.6, -2.3, 3.7 };`
`for (double x : a) ma.update(x);`
`cout << fixed << setprecision(2) <<  ma.get() << '\n';`
	`2.60`

Resposta: (regime de penalização: 0, 0, 0, 0, 10, 20, 30, ... %)

Limpar resposta

```
// Submit your code using file attachments!
```

MovingAverage.cpp

	Teste	Esperado	Recebido	

✓	`MovingAverage ma(1);`
	`const MovingAverage& r = ma;`
	`ma.update(1.0);`
	`cout << fixed << setprecision(2) << r.get() << '\n';`
		`1.00`	`1.00`	✓

✓	`MovingAverage ma(1);`
	`double a[] { 1.2, 3.4, 5.0 };`
	`for (double x : a) {`
	`  ma.update(x);`
	`  cout << fixed << setprecision(2) <<  ma.get() << ' ';`
	`}`
	`cout << fixed << setprecision(2) <<  ma.get() << '\n';`
		`1.20 3.40 5.00 5.00`	`1.20 3.40 5.00 5.00`	✓

✓	`MovingAverage ma(2);`
	`double a[] { 1.2, 3.4, 5.0 };`
	`for (double x : a) {`
	`  ma.update(x);`
	`  cout << fixed << setprecision(2) <<  ma.get() << ' ';`
	`}`
	`cout << fixed << setprecision(2) <<  ma.get() << '\n';`
		`1.20 2.30 4.20 4.20`	`1.20 2.30 4.20 4.20`	✓

✓	`MovingAverage ma(3);`
	`double a[] { 1.2, 3.4, 5.0 };`
	`for (double x : a) {`
	`  ma.update(x);`
	`  cout << fixed << setprecision(2) <<  ma.get() << ' ';`
	`}`
	`cout << fixed << setprecision(2) <<  ma.get() << '\n';`
		`1.20 2.30 3.20 3.20`	`1.20 2.30 3.20 3.20`	✓

✓	`MovingAverage ma(4);`
	`double a[] { 0.1, 1.2, -1.2, 3.4, 5.6, -2.3, 3.7 };`
	`for (double x : a) ma.update(x);`
	`cout << fixed << setprecision(2) <<  ma.get() << '\n';`
		`2.60`	`2.60`	✓

Passou em todos os testes! ✓

Solução do autor da pergunta (C):

```
/*
  // private tests (1000 points each)
```

Correta

Nota desta submissão: 20/20

Pergunta 4

Correta Pontuou 20 de 20

Write the C++ code for two template functions `to_string` and `replace`, declared as:

```
template <typename Itr>
string to_string(Itr start, Itr end)

template <typename Itr, typename T>
int replace(Itr start, Itr end, const T& a, const T& b)
```

such that:

- `to_string(start, end)` takes two iterators `start` and `end` associated with the same container and yields a string representation of the form `"[ elem_1 elem_2 ... elem_n ]"` of all elements of the container between `start` and `end`, including `start` but excluding `end`; and
- `replace(start, end, a, b)` traverses all elements between `start` and `end`, including `start` but excluding `end`, replacing values of `a` by values of `b`, and returns the number of elements that were replaced.

Hints:

- Consider the use of the following iterator operators: `++`, `!=` and `*`.
- Use an `ostringstream` object in `to_string()` to build the string result.

Por exemplo:

Teste	Resultado

`vector<int> v;`
`cout << replace(v.begin(), v.end(), 0, 1) << ' '`
`     << to_string(v.cbegin(), v.cend()) << '\n';`
	`0 [ ]`

`vector<int> v { 1, 2, 3, 3, 4 };`
`cout << replace(v.begin(), v.end(), 3, 0) << ' '`
`     << to_string(v.cbegin(), v.cend()) << '\n';`
	`2 [ 1 2 0 0 4 ]`

`vector<int> v { 5, 1, 2, 3, 4, 5 };`
`cout << replace(v.begin() + 1, v.end() - 1, 5, 0) << ' '`
`     << to_string(v.cbegin(), v.cend()) << '\n';`
	`0 [ 5 1 2 3 4 5 ]`

`list<string> l { "C++", "Java", "C++", "Python", "Rust", "C" } ;`
`cout << replace(l.begin(), l.end(), string("C++"), string("Rust")) << ' '`
`     << replace(l.begin(), l.end(), string("C"), string("Rust")) << ' '`
`     << to_string(l.cbegin(), l.cend()) << '\n';`
	`2 1 [ Rust Java Rust Python Rust Rust ]`

`string s = "Hello world";`
`cout << replace(s.begin(), s.end(), 'l', 'L') << ' '`
`     << to_string(s.cbegin(), s.cend()) << ' '`
`     << replace(s.rbegin(), s.rend(), 'o', 'O') << ' '`
`     << to_string(s.crbegin(), s.crend()) << ' '`
`     << to_string(s.cbegin(), s.cend()) << '\n';`
	`3 [ H e L L o   w o r L d ] 2 [ d L r O w   O L L e H ] [ H e L L O   w O r L d ]`

Resposta: (regime de penalização: 0, 0, 0, 0, 10, 20, 30, ... %)

Limpar resposta

```
#include <vector>
#include <list>
#include <iostream>
#include <sstream>

using namespace std;

template <typename Itr>
string to_string(Itr start, Itr end){
    ostringstream oss;
    oss << "[ ";
    while(start != end){
        oss<< *start << " ";
        start++;
    }
    oss << "]";
    return oss.str();
}

template <typename Itr, typename T>
int replace(Itr start, Itr end, const T& a, const T& b){
    int count = 0;
```

	Teste	Esperado	Recebido	

✓	`vector<int> v;`
	`cout << replace(v.begin(), v.end(), 0, 1) << ' '`
	`     << to_string(v.cbegin(), v.cend()) << '\n';`
		`0 [ ]`	`0 [ ]`	✓

✓	`vector<int> v { 1, 2, 3, 3, 4 };`
	`cout << replace(v.begin(), v.end(), 3, 0) << ' '`
	`     << to_string(v.cbegin(), v.cend()) << '\n';`
		`2 [ 1 2 0 0 4 ]`	`2 [ 1 2 0 0 4 ]`	✓

✓	`vector<int> v { 5, 1, 2, 3, 4, 5 };`
	`cout << replace(v.begin() + 1, v.end() - 1, 5, 0) << ' '`
	`     << to_string(v.cbegin(), v.cend()) << '\n';`
		`0 [ 5 1 2 3 4 5 ]`	`0 [ 5 1 2 3 4 5 ]`	✓

✓	`list<string> l { "C++", "Java", "C++", "Python", "Rust", "C" };`
	`cout << replace(l.begin(), l.end(), string("C++"), string("Rust")) << ' '`
	`     << replace(l.begin(), l.end(), string("C"), string("Rust")) << ' '`
	`     << to_string(l.cbegin(), l.cend()) << '\n';`
		`2 1 [ Rust Java Rust Python Rust Rust ]`	`2 1 [ Rust Java Rust Python Rust Rust ]`	✓

✓	`string s = "Hello world";`
	`cout << replace(s.begin(), s.end(), 'l', 'L') << ' '`
	`     << to_string(s.cbegin(), s.cend()) << ' '`
	`     << replace(s.rbegin(), s.rend(), 'o', 'O') << ' '`
	`     << to_string(s.crbegin(), s.crend()) <<  ' '`
	`     << to_string(s.cbegin(), s.cend()) << '\n';`
		`3 [ H e L L o   w o r L d ] 2 [ d L r O w   O L L e H ] [ H e L L O   w O r L d ]`	`3 [ H e L L o   w o r L d ] 2 [ d L r O w   O L L e H ] [ H e L L O   w O r L d ]`	✓

Passou em todos os testes! ✓

Solução do autor da pergunta (C):

```
#include <vector>
#include <list>
#include <iostream>
#include <sstream>
using namespace std;

template <typename Itr>
string to_string(Itr start, Itr end) {
   ostringstream out;
   Itr itr = start;
   out << '[';
   while (itr != end) {
      out << ' ' << *itr;
      itr++;
   }
   out << " ]";
```

```
    return out.str();
}

template <typename Itr, typename T>
size_t replace(Itr start, Itr end, const T& a , const T& b) {
    Itr itr = start;
```

Correta

Nota desta submissão: 20/20

Pergunta 5

Correta Pontuou 20 de 20

Consider the partial definitions of classes `Page` and `Book` in the files `Page.h` and `Book.h` and the partial implementation of the classes in the files `Page.cpp` and `Book.cpp` given in the archive `ex5.zip`.

Write the C++ code of the member function `void Book::build_index(const set<string>& words)` that builds the index of a book.

- The words to be indexed are received through parameter `words`
- The index of the book is a `map` that associates to each word a `set` containing the numbers of the pages in which the word occurs (see the private data members of class `Book`)
- If there are no occurrences of some words present in the `words` parameter they must be omitted from the index
- Each book is saved in a text file, named `book_xx.txt` where `xx` takes the values "01", "02", ..."05", available in the given archive; the string `<PAGE>` marks the end of each page (see the given examples)
- For simplicity, consider that the book does not contain punctuation marks
- The examples used for testing are not real books but just collections of words, organised in pages

You must submit only the code for function `Book::build_index`.

In your local workspace, to compile a program in a file containing the tests, named for instance `main.cpp`, you should use the supplied `Makefile` and execute the following command in the Terminal:

```
make PROG=main CPP_FILES="Page.cpp Book.cpp main.cpp" HEADERS="Page.h Book.h"
```

Por exemplo:

Teste	Resultado

`Book b;`
`if (!(b.import("book_01.txt"))) cout << "book not found!\n";`
`set<string> words = { "C++", "linux", "computer" };`
`b.build_index(words);`
`b.show_index();`
	`/ C++: 1 / linux: 1 /`

`Book b;`
`if (!(b.import("book_02.txt"))) cout << "book not found!\n";`
`set<string> words = { "C++", "program", "computer", "windows" };`
`b.build_index(words);`
`b.show_index();`
	`/ C++: 1 2 3 / computer: 3 / program: 1 / windows: 1 3 /`

`Book b;`
`if (!(b.import("book_03.txt"))) cout << "book not found!\n";`
`set<string> words = { "linux", "program", "cpu" };`
`b.build_index(words);`
`b.show_index();`
	`/ cpu: 1 / linux: 2 / program: 2 /`

`Book b;`
`if (!(b.import("book_04.txt"))) cout << "book not found!\n";`
`set<string> words = { "computer", "program", "file","windows" };`
`b.build_index(words);`
`b.show_index();`
	`/ computer: 1 / file: 3 / program: 1 3 / windows: 3 /`

`Book b;`
`if (!(b.import("book_05.txt"))) cout << "book not found!\n";`
`set<string> words = { "linux", "byte", "cpu" };`
`b.build_index(words);`
`b.show_index();`
	`/ byte: 1 2 / cpu: 1 /`

Resposta: (regime de penalização: 0, 0, 0, 0, 10, 20, 30, ... %)

Limpar resposta

```
#include <iostream>
#include <sstream>
#include <string>
#include <set>
#include "Page.h"
#include "Book.h"
```

```

using namespace std;

//! Builds "index" for the "words" received as parameter
void Book::build_index(const set<string>& words) {
    for (string target : words) {
        for (size_t i = 0; i < book_.size(); i++) {
            Page page = book_[i];
            for (size_t j = 0; j < page.get_num_lines(); j++) {
                istringstream line(page.get_line(j));
                string word;
                while (line >> word) {
                    if (word == target) {
                        index_[target].insert(i + 1);
                    }
                }
```

	Teste	Esperado	Recebido	

✓	`Book b;`
	`if (!(b.import("book_01.txt"))) cout << "book not found! \n";`
	`set<string> words = { "C++", "linux", "computer" };`
	`b.build_index(words);`
	`b.show_index();`
		`/ C++: 1 / linux: 1 /`	`/ C++: 1 / linux: 1 /`	✓

✓	`Book b;`
	`if (!(b.import("book_02.txt"))) cout << "book not found! \n";`
	`set<string> words = { "C++", "program", "computer", "windows" };`
	`b.build_index(words);`
	`b.show_index();`
		`/ C++: 1 2 3 / computer: 3 / program: 1 / windows: 1 3 /`	`/ C++: 1 2 3 / computer: 3 / program: 1 / windows: 1 3 /`	✓

✓	`Book b;`
	`if (!(b.import("book_03.txt"))) cout << "book not found! \n";`
	`set<string> words = { "linux", "program", "cpu" };`
	`b.build_index(words);`
	`b.show_index();`
		`/ cpu: 1 / linux: 2 / program: 2 /`	`/ cpu: 1 / linux: 2 / program: 2 /`	✓

✓	`Book b;`
	`if (!(b.import("book_04.txt"))) cout << "book not found! \n";`
	`set<string> words = { "computer", "program", "file","windows" };`
	`b.build_index(words);`
	`b.show_index();`
		`/ computer: 1 / file: 3 / program: 1 3 / windows: 3 /`	`/ computer: 1 / file: 3 / program: 1 3 / windows: 3 /`	✓

✓	`Book b;`
	`if (!(b.import("book_05.txt"))) cout << "book not found! \n";`
	`set<string> words = { "linux", "byte", "cpu" };`
	`b.build_index(words);`
	`b.show_index();`
		`/ byte: 1 2 / cpu: 1 /`	`/ byte: 1 2 / cpu: 1 /`	✓

Passou em todos os testes! ✓

Solução do autor da pergunta (C):

```
#include <iostream>
#include <sstream>
#include <string>
#include <set>
#include "Page.h"
#include "Book.h"

using namespace std;

//! Builds ‘index’ for the ‘words’ received as parameter
void Book::build_index(const set<string>& words) {
  size_t page_num = 0;
  index_.clear();

  for (const Page& page : book_) {
    ++page_num;
    for (size_t i = 0; i < page.get_num_lines(); ++i) {
      string line = page.get_line(i);
      string word;
      istringstream iss(line);
      while (iss >> word) {  // extract words from line
        if (words.find(word) != words.end()) { // 'word' must be indexed
```

Correta

Nota desta submissão: 20/20

◀ T09 17/05

Ir para...

T10 24/05 ▶