ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук

Департамент программной инженерии

Пояснительная записка к домашнему заданию

по дисциплине «Архитектура вычислительных систем»

Разработка многопоточных приложений с использованием OpenMP.

Исполнитель:

студент группы

БПИ193(1)

Полянская П. А.

Вариант 17

01.12.2020 г.

Текст задачи.

Задача об инвентаризации по книгам. После нового года в библиотеке университета обнаружилась пропажа каталога. После поиска и наказания, виноватых ректор дал указание восстановить каталог силами студентов. Фонд библиотека представляет собой прямоугольное помещение, в котором находится M рядов по N шкафов по K книг в каждом шкафу. Требуется создать многопоточное приложение, составляющее каталог. При решении задачи использовать метод «портфель задач», причем в качестве отдельной задачи задается внесение в каталог записи об отдельной книге.

О программе.

Используется метод “портфель задач” (задан по условию).

Задача – внести в каталог книгу, зная её название и порядковый номер.

Названия книг из списка состоят из слова Book и порядкового номера – это номер книги в списке после добавления, так как названия книг создаю я и они идут по порядку, то можно считать их уже отсортированными.

OpenMP - используется секция для распараллеливания цикла for.

Используемые источники.

1) Параллельное программирование на OpenMP. [Электронный ресурс]

http://ccfit.nsu.ru/arom/data/openmp.pdf (дата обращения: 01.12.2020)

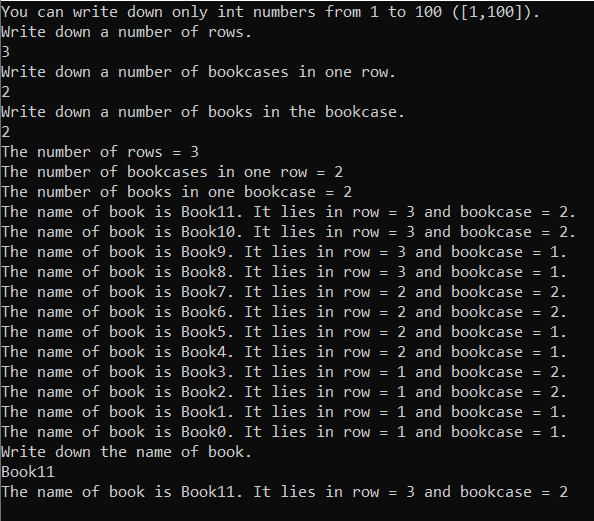
2) SoftCraft, сайт по учебной дисциплине. [Электронный ресурс]

http://softcraft.ru/ (дата обращения: 01.12.2020)

Пояснения по решению.

Пояснения присутствуют в коде, и есть комментарии о роли каждого метода.

Тестирование различных входных данных.

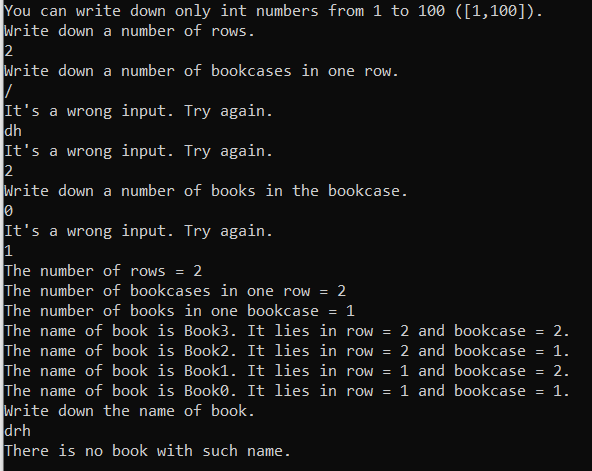
1. Входные данные правильные.
2. Введены неправильные входные данные.

Были попробованы введены строка(буквы), дробное число, отрицательное число, число равное 0 и больше 100.

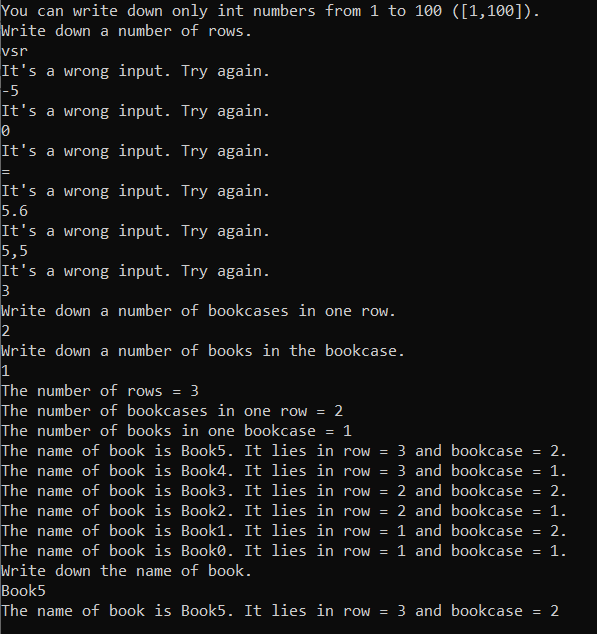
Все неправильные значения были обработаны и выведена строка о неправильном введённом значении.

Пользователь вводит число, пока оно не окажется правильным.

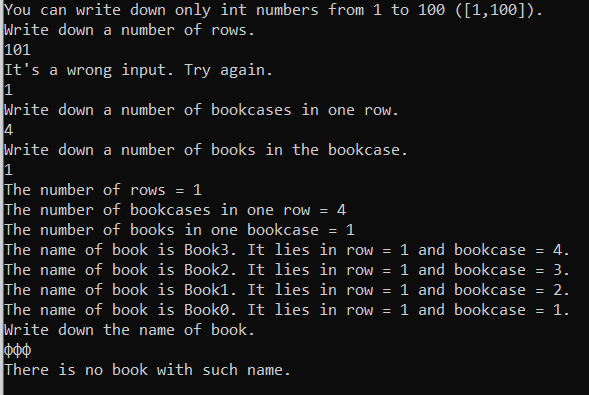
2.1)



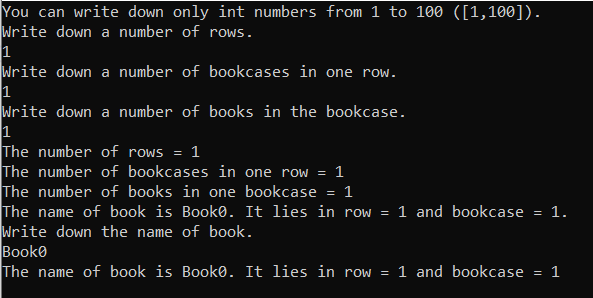
2.2)



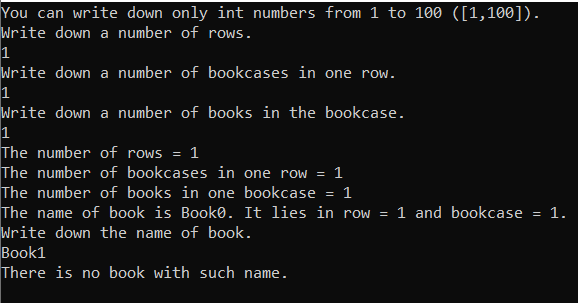
2.3)



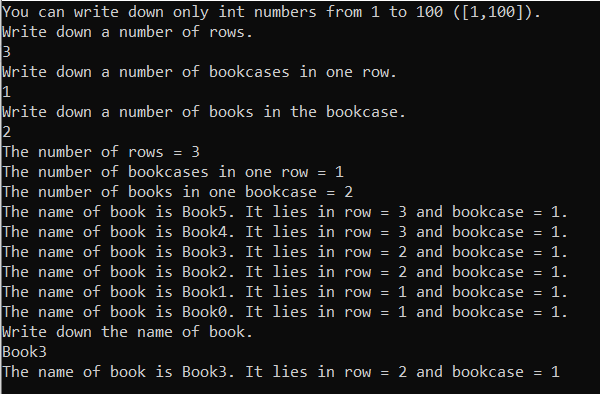
3)



4)



5)



Исходный код.

#include <iostream>

#include <fstream>

#include <cstdlib>

#include <stdlib.h>

#include <string>

#include <vector>

using namespace std;

//Полянская Полина Алексеевна БПИ193 17 вариант

/\*

Условие:

Задача об инвентаризации по книгам. После нового года в библиотеке университета обнаружилась пропажа каталога.

После поиска и наказания, виноватых ректор дал указание восстановить каталог силами студентов. Фонд библиотека представляет собой прямоугольное помещение, в котором находится M рядов по N шкафов по K книг в каждом шкафу.

Требуется создать многопоточное приложение, составляющее каталог. При решении задачи использовать метод «портфель задач», причем в качестве отдельной задачи задается внесение в каталог записи об отдельной книге.

\*/

static vector <pair<pair<int, int>, string>> library;

static vector<pair<string, int>> namesAndNums;

static int M, N, K = 0;

static int Length = 0;

/// <summary>

/// Проверяет, удовлетворяет ли введенное число всем условиям, и превращает строку в число

/// </summary>

/// <param name="str">Строка о том, какое число вводит пользователь</param>

/// <returns>Возвращает правильное число</returns>

int Converter(string str)

{

cout << str << endl;

bool Bool = true;

string check;

int res = 0;

while (Bool)

{

try {

cin >> check;

res = stoi(check);

//проверяет если число <=0, больше 100 или не целое, то повторяется ввод строки

if (res <= 0 || to\_string(res) != check || res > 100)

{

throw exception();

}

Bool = false;

}

catch (exception e) {

cout << "It's a wrong input. Try again." << endl;

}

}

return res;

}

/// <summary>

/// Добавляет название книги и её номер в вектор

/// </summary>

/// <param name="i">порядковый номер</param>

void addNamesAndNums(int i)

{

string numToSTr = to\_string(i);

string s = "Book" + numToSTr;

namesAndNums.push\_back(make\_pair(s, i));

}

/// <summary>

/// Берёт элемент из вектора books\_names и добавляет в вектор library элемент, после удаляет этот элемент из books\_names.

/// </summary>

void full\_catalog()

{

vector<pair<string, int>> name\_num\_of\_str;

name\_num\_of\_str.push\_back(namesAndNums[namesAndNums.size() - 1]);

auto it = name\_num\_of\_str.begin();

vector<int> nums(2);

int number\_in\_vector = it->second + 1;

//Узнаём, в каком ряду находится данная книга

for (int j = 0; j < M; j++)

{

if (number\_in\_vector > j \* N \* K && number\_in\_vector <= (j + 1) \* N \* K)

{

nums[0] = j + 1;

}

}

//Узнаём, в каком шкафу находится данная книга

int num\_left = number\_in\_vector - (nums[0] - 1) \* N \* K;

for (int j = 0; j < N; j++)

{

if (num\_left > j \* K && num\_left <= (j + 1) \* K)

{

nums[1] = j + 1;

}

}

string name = it->first;

library.push\_back(make\_pair(make\_pair(nums[0], nums[1]), name));

//Удаляем добавленную в каталог книгу

namesAndNums.pop\_back();

}

int main()

{

cout << "You can write down only int numbers from 1 to 100 ([1,100])." << endl;

//Пользователь вводит число рядов

M = Converter("Write down a number of rows.");

//Пользователь вводит число шкафов в ряду

N = Converter("Write down a number of bookcases in one row.");

//Пользователь вводит число книг в одном шкафу

K = Converter("Write down a number of books in the bookcase.");

//Вывод установленных значений

cout << "The number of rows = " << M << endl;

cout << "The number of bookcases in one row = " << N << endl;

cout << "The number of books in one bookcase = " << K << endl;

vector<int> numbers = { M,N,K };

int amount\_of\_books = M \* N \* K;

for (int i = 0; i < amount\_of\_books; i++)

{

addNamesAndNums(i);

}

Length = namesAndNums.size();

#pragma omp parallel

{

#pragma omp for

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

full\_catalog();

}

auto it = library.begin();

//Выводятся положения всех книг

while (it != library.end())

{

int m = it->first.first;

int n = it->first.second;

string name = it->second;

cout<<"The name of book is "<<name<<". It lies in row = "<< m<< " and bookcase = "<<n<<"."<<endl;

it++;

}

//Ввод названия книги для поиска

cout << "Write down the name of book." << endl;

string str;

bool checker = false;

cin >> str;

it = library.begin();

//Если книга с данным названием найдется, то выводятся её координаты

while (it != library.end())

{

int m = it->first.first;

int n = it->first.second;

string name = it->second;

if (name.\_Equal(str))

{

cout << "The name of book is " << name << ". It lies in row = " << m << " and bookcase = " << n << endl;

checker = true;

}

it++;

}

//Если такой книги не нашлось, то выводится информация об этом.

if (!checker)

{

cout << "There is no book with such name." << endl;

}

}