Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №7

за 4 семестр

По дисциплине: «ООТПиСП»

Тема: «Умные указатели»

Выполнил:

Студент 2 курса

Группы ПО-4(1)

Иваненко И. Л.

Проверил:

Миндер А. В.

2021

Лабораторная работа №7

Умные указатели

Цель работы: получить практические навыки работы с умными указателями.

Вариант 10

|  |  |  |  |
| --- | --- | --- | --- |
| 10 | Удалить все элементы с нечетными индексами | Добавить К столбцов в начало матрицы | Задана символьная матрица размером NxM. Определить количество различных элементов матрицы (т.е. повторяющиеся элементы считать один раз). |

Код программы:

task1.h

#ifndef SRC\_TASK1\_H  
#define SRC\_TASK1\_H  
  
void task1();  
  
#endif //SRC\_TASK1\_H

task1.cpp

#include "task1.h"  
#include <cmath>  
#include <iostream>  
#include <memory>  
  
void task1() {  
 size\_t size = 15;  
 std::unique\_ptr<int []> arr = std::make\_unique<int []>(size);  
 for(int i = 0; i < size; i++) {  
 arr[i] = i;  
 }  
  
 std::cout << "\nARRAY\n";  
 for(size\_t i = 0; i < size; i++) {  
 std::cout << arr[i] << " ";  
 }  
  
 size\_t new\_arr\_size = std::ceil((static\_cast<float>(size) / 2));  
 std::unique\_ptr<int []> new\_arr = std::make\_unique<int []>(new\_arr\_size);  
  
 for(size\_t i = 0, j = 0; i < size; i += 2) {  
 new\_arr[j] = arr[i];  
 j++;  
 }  
  
 arr = std::move(new\_arr);  
  
 std::cout << "\nRESULT\n";  
  
 for(size\_t i = 0; i < new\_arr\_size; i++) {  
 std::cout << arr[i] << " ";  
 }  
}

task2.h

#ifndef SRC\_TASK2\_H  
#define SRC\_TASK2\_H  
  
void task2();  
  
#endif //SRC\_TASK2\_H

task2.cpp

#include <iostream>  
#include <memory>  
  
void task2() {  
 size\_t N = 5;  
 std::unique\_ptr<std::unique\_ptr<int[]>[]> matrix(std::make\_unique<std::unique\_ptr<int[]>[]>(N));  
 for(size\_t i = 0; i < N; i++) {  
 matrix[i] = std::make\_unique<int[]>(N);  
 }  
  
 for(size\_t i = 0; i < N; i++) {  
 for(size\_t j = 0; j < N; j++) {  
 matrix[i][j] = i + j;  
 }  
 }  
  
 std::cout << "\nMATRIX\n";  
 for(size\_t i = 0; i < N; i++) {  
 for(size\_t j = 0; j < N; j++) {  
 std::cout << matrix[i][j] << " ";  
 }  
 std::cout << std::endl;  
 }  
  
 size\_t K = 3;  
 std::unique\_ptr<std::unique\_ptr<int[]>[]> temp\_matrix(std::make\_unique<std::unique\_ptr<int[]>[]>(N));  
 for(size\_t i = 0; i < N; i++) {  
 temp\_matrix[i] = std::make\_unique<int[]>(N + K);  
 }  
  
 for(size\_t i = 0; i < N; i++) {  
 for(size\_t j = K; j < N + K; j++) {  
 temp\_matrix[i][j] = matrix[i][j - K];  
 }  
 }  
  
 for(size\_t i = 0; i < N; i++) {  
 for(size\_t j = 0; j < K; j++) {  
 temp\_matrix[i][j] = 1;  
 }  
 }  
  
 matrix = std::move(temp\_matrix);  
  
 std::cout << "\nRESULT\n";  
  
 for(size\_t i = 0; i < N; i++) {  
 for(size\_t j = 0; j < N + K; j++) {  
 std::cout << matrix[i][j] << " ";  
 }  
 std::cout << std::endl;  
 }  
}

task3.h

#ifndef SRC\_TASK3\_H  
#define SRC\_TASK3\_H  
  
void task3();  
  
#endif //SRC\_TASK3\_H

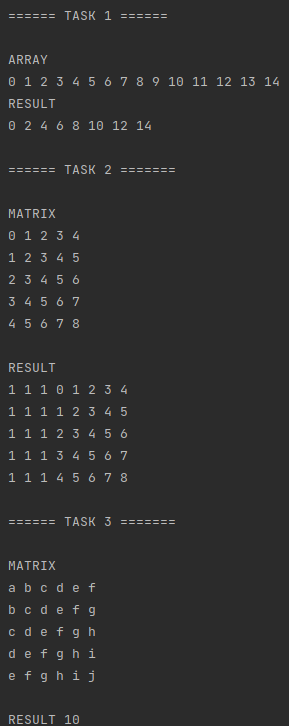
task3.cpp

#include <iostream>  
#include <memory>  
#include <map>  
  
void task3() {  
 size\_t N = 5;  
 size\_t M = 6;  
  
 std::unique\_ptr<std::unique\_ptr<char[]>[]> matrix(std::make\_unique<std::unique\_ptr<char[]>[]>(N));  
 for(size\_t i = 0; i < N; i++) {  
 matrix[i] = std::make\_unique<char[]>(M);  
 }  
  
 for(size\_t i = 0; i < N; i++) {  
 for(size\_t j = 0; j < M; j++) {  
 matrix[i][j] = 'a' + i + j;  
 }  
 }  
  
 std::cout << "\nMATRIX\n";  
  
 for(size\_t i = 0; i < N; i++) {  
 for(size\_t j = 0; j < M; j++) {  
 std::cout << matrix[i][j] << " ";  
 }  
 std::cout << std::endl;  
 }  
  
 std::map<char, size\_t> chars\_frequency;  
 size\_t unique\_chars = 0;  
  
 for(size\_t i = 0; i < N; i++) {  
 for(size\_t j = 0; j < M; j++) {  
 if(chars\_frequency.count(matrix[i][j]) == 0) {  
 chars\_frequency[matrix[i][j]] = 0;  
 unique\_chars++;  
 } else {  
 chars\_frequency[matrix[i][j]]++;  
 }  
 }  
 }  
  
 std::cout << "\nRESULT " << unique\_chars;  
}

main.cpp

#include <iostream>  
#include "task1.h"  
#include "task2.h"  
#include "task3.h"  
  
int main() {  
 std::cout << "\n====== TASK 1 ======\n";  
 task1();  
 std::cout << "\n\n====== TASK 2 =======\n";  
 task2();  
 std::cout << "\n====== TASK 3 =======\n";  
 task3();  
 return 0;  
}

Результат выполнения:



Вывод: В ходе выполнения данной лабораторной работы ознакомился с работой с умными указателями.