***Министерство образования Республики Беларусь***

***Учреждение Образования***

***«Брестский Государственный Технический Университет»***

***Кафедра ИИТ***

**Лабораторная работа №2**

**По дисциплине ОАиП за II семестр**

**Тема: «Составные типы данных в процедурах и функциях»**

**Выполнил:**

Студент 1-го курса

Группы ИИ-18

Смоляков А.Д.

**Проверил:**

Михно Е. В.

Брест 2020

**Лабораторная работа №2**

Цель: научиться работать с указателями на функции.

**Вариант 15**

Задание 1. Описать процедуру Split2(A, NA, B, NB, C, NC), формирующую по целочисленному массиву A размера NA два целочисленных массива B и C размера NB и NC соответственно; при этом массив B содержит все четные числа из массива A, а массив C — все нечетные числа (в том же порядке). Массивы B и C и числа NB и NC являются выходными параметрами. Применить эту процедуру к данному массиву A размера NA и вывести размер и содержимое полученных массивов B и C.

Код программы:

#include <iostream>

#include <ctime>

using namespace std;

int Split2(const int \*in\_arr, const int in\_n, int \*& b, int \*& c)

{

int even\_n = 0, odd\_n = 0;

for (int i = 0; i < in\_n; i++)

{

if (!(in\_arr[i] % 2))

++even\_n;

else

++odd\_n;

}

b = new int[even\_n];

c = new int[odd\_n];

even\_n = 0; odd\_n = 0;

for (int i = 0; i < in\_n; i++)

{

if (!(in\_arr[i] % 2))

b[even\_n++] = in\_arr[i];

else

c[odd\_n++] = in\_arr[i];

}

return even\_n;

}

int main()

{

int n;

cout << "input the number of array elements: ";

cin >> n;

int \*arr = new int[n];

srand(time(0));

cout << "source array: ";

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

cout << (arr[i] = rand() % 101) << " ";

cout << endl;

int \*arr\_even,

\*arr\_odd,

even\_n,

odd\_n;

even\_n = Split2(arr, n, arr\_even, arr\_odd);

odd\_n = n - even\_n;

cout << "array of even elements from source: ";

for (int i = 0; i < even\_n; i++)

cout << arr\_even[i] << " ";

cout << endl;

cout << "array of odd elements from source: ";

for (int i = 0; i < odd\_n; i++)

cout << arr\_odd[i] << " ";

cout << endl;

delete[] arr;

delete[] arr\_even;

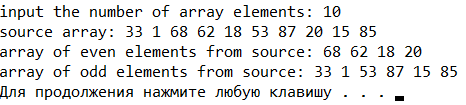
delete[] arr\_odd;

system("pause");

return 0;

}

Вывод:



Задание 2. Описать функцию DecToBin(N) строкового типа, возвращающую строковое представление целого неотрицательного числа N в двоичной системе счисления. Результирующая строка состоит из символов «0»–«1» и не содержит ведущих нулей (за исключением представления числа 0).Используя эту функцию, получить двоичные представления пяти данных чисел.

Код программы:

#include <iostream>

#include <algorithm>

#include <string>

using namespace std;

string dec\_to\_bin(unsigned int n)

{

if (n == 0)

return "0";

string res;

do

{

res.push\_back(n % 2 ? '1' : '0');

n /= 2;

} while (n != 0);

reverse(res.begin(), res.end());

return res;

}

int main()

{

const int cnt = 5;

int val;

for (int i = 1; i <= cnt; ++i)

{

cout << "[" << i << "\\" << cnt << "]: ";

cin >> val;

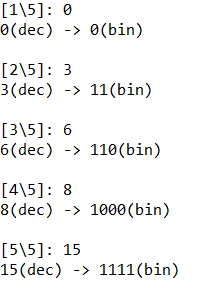
cout << val << "(dec) -> " << dec\_to\_bin(val) << "(bin)\n" << std::endl;

}

system("pause");

}

Вывод:



Задание 3. Описать рекурсивную функцию Fib2(N) целого типа, вычисляющую N-й элемент последовательности чисел Фибоначчи (N — целое число):

F1 = F2 = 1, FK = FK–2 + FK–1, K = 3, 4, … .

Считать, что номер N не превосходит 20. Для уменьшения количества рекурсивных вызовов по сравнению с функцией Fib1 (см. задание 4) создать вспомогательный массив для хранения уже вычисленных чисел Фибоначчи и обращаться к нему при выполнении функции Fib2. С помощью функции Fib2 найти пять чисел Фибоначчи с данными номерами.

Код программы:

#include <Windows.h>

#include <iostream>

using namespace std;

const int notName = system("color f0");

int Fib2(int, int[]);

int main()

{

int \* arr = new int[20];

arr[0] = arr[1] = 1;

int \* res = new int[5];

for (int i = 0; i < 5; i++)

{

int num;

cout << "\n Number: "; cin >> num;

res[i] = Fib2(num, arr);

cout << " Fib2: " << res[i] << endl;

}

cout << endl;

system("pause");

return 0;

}

int Fib2(int N, int arr[])

{

if (N > 2)

{

Fib2(N - 1, arr);

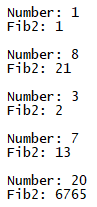
arr[N - 1] = arr[N - 2] + arr[N - 3];

}

return arr[N - 1];

}

Вывод:



Вывод: по ходу лабораторной работы научился работать с указателями на функции.