МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

Теоретико-множественные основы интеллектуальных систем

ОТЧЁТ

По лабораторной работе №2

Выполнил:

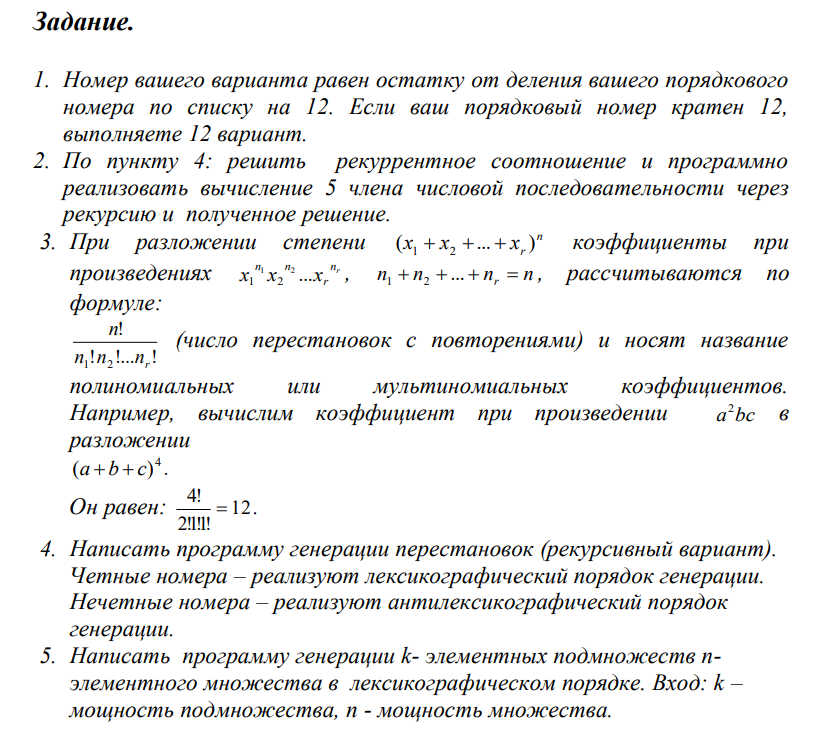
Студент группы ИИ-22

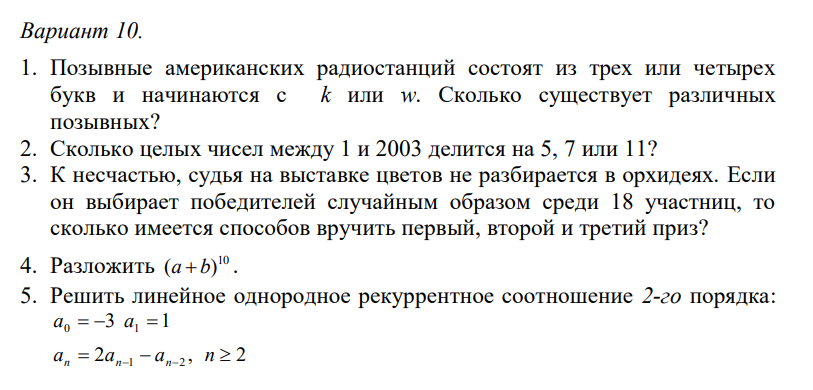
Копанчук Евгений Романович

Проверил:

Глущенко Т. А.

Брест - 2022





***Файл Main.cpp***

#include "Header.h"

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

std::cout << std::endl << "Вывод пяти членов заданной рекурсии: " << std::endl;

std::cout << -3 << " " << 1 << " ";

recursion1(1, -3, 3);

std::cout << std::endl << "Вывод пяти членов полученной рекурсии: " << std::endl;

recursion2(0, 4);

std::cout << std::endl << std::endl << "Генерация перестановок в лексикографической порядке: " << std::endl;

int\* a, Num = 1;

a = new int[3];

for (int i = 0; i < 3; i++)

a[i] = i + 1;

std::cout << std::setw(3) << Num++ << ": ";

Print(a, 3);

while (NextSet(a, 3)) {

std::cout << std::setw(3) << Num++ << ": ";

Print(a, 3);

}

std::cout << std::endl << "Подмножества, состоящие из 2 элементов, множества 4 элементов: " << std::endl;

all\_combinations(4, 2);

std::cout << std::endl;

}

***Файл Source.cpp***

#include "Header.h"

// Задание 2 Рекурсия

int recursion1(int fst, int snd, int k) {

int a = 2 \* fst - snd;

std::cout << a << " ";

if (k != 1) {

k--;

return recursion1(a, fst, k);

}

}

int recursion2(int n, int k) {

std::cout << -3 + 4 \* n << " ";

if (n != k) {

n++;

return recursion2(n, k);

}

}

// Задание 4 Перестановки

void swap(int\* a, int i, int j) {

int s = a[i];

a[i] = a[j];

a[j] = s;

}

bool NextSet(int\* a, int n) {

int j = n - 2;

while (j != -1 && a[j] >= a[j + 1]) j--;

if (j == -1)

return false;

int k = n - 1;

while (a[j] >= a[k]) k--;

swap(a, j, k);

int l = j + 1, r = n - 1;

while (l < r)

swap(a, l++, r--);

return true;

}

void Print(int\* a, int n) {

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

std::cout << a[i] << " ";

std::cout << std::endl;

}

// Задание 5 Подмножества

int gray\_code(int n) {

return n ^ (n >> 1);

}

int count\_bits(int n) {

int res = 0;

for (; n; n >>= 1)

res += n & 1;

return res;

}

void all\_combinations(int n, int k) {

int Num = 0;

for (int i = 0; i < (1 << n); ++i) {

int cur = gray\_code(i);

if (count\_bits(cur) == k) {

Num++;

std::cout << std::endl << std::setw(3) << Num << ":";

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

if (cur & (1 << j))

std::cout << " " << j + 1;

}

}

}

***Файл Header.h***

#pragma once

#include <iostream>

#include <cmath>

#include <iomanip>

#include <Windows.h>

#include <algorithm>

// Задание 2 Рекурсия

int recursion1(int fst, int snd, int k);

int recursion2(int n, int k);

// Задание 4 Перестановки

void Print(int\* a, int n);

bool NextSet(int\* a, int n);

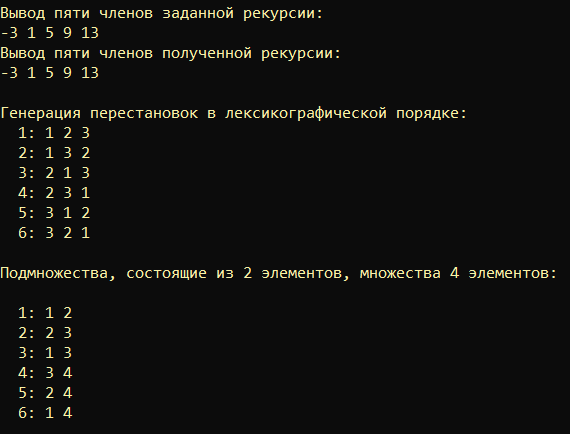
// Задание 5 Подмножества

int gray\_code(int n);

int count\_bits(int n);

void all\_combinations(int n, int k);

***Вывод программы***

******