Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ»

Отчёт по лабораторной работе №2

«**Комбинаторные алгоритмы решения**

**оптимизационных задач**»

Студентка: Змитревич Д.А.

ФИТ 2 курс 9 группа

Преподаватель: Барковский Е.В.

Минск 2022

**ЦЕЛЬ РАБОТЫ:** приобрести навыки разработки генераторов подмножеств, перестановок, сочетаний и размещений на С++; научиться применять разработанные генераторы для решения задач о рюкзаке (упрощенную, коммивояжера, об оптимальной загрузке судна и об оптимальной загрузке судна с центровкой.

**Задание 1**. Разработать генератор подмножеств заданного множества,

файл **LB2\_1.cpp**, содержащий вызов функций:

#include <iostream>

#include "Combi1.h"

#include <tchar.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

setlocale(LC\_ALL, "rus");

char AA[][2] = { "A", "B", "C", "D" };

std::cout << " -- Генератор множества всех подмножеств --";

std::cout << std::endl << "Исходное множество: ";

std::cout << "{ ";

for (int i = 0; i < sizeof(AA) / 2; i++)

std::cout << AA[i] << ((i < sizeof(AA) / 2 - 1) ? ", " : " ");

std::cout << "}";

std::cout << std::endl << "Генерация всех подмножеств ";

combi1::subset s1(sizeof(AA) / 2); // создание генератора

int n = s1.getfirst(); // первое (пустое) подмножество

while (n >= 0) // пока есть подмножества

{

std::cout << std::endl << "{ ";

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

std::cout << AA[s1.ntx(i)] << ((i < n - 1) ? ", " : " ");

std::cout << "}";

n = s1.getnext(); // cледующее подмножество

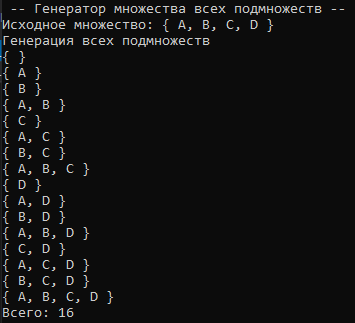
};

std::cout << std::endl << "Всего: " << s1.count() << std::endl;

system("pause");

return 0;

}



**Задание 2**. Задание 2. Разработать генератор сочетаний,

файл **LB2\_2.cpp**, содержащий вызов функций:

#include <iostream>

#include <tchar.h>

#include "Combi2.h"

int \_tmain(int argc, \_TCHAR\* argv[])

{

setlocale(LC\_ALL, "rus");

char AA[][2] = { "A", "B", "C", "D", "E" };

std::cout << std::endl << " -- Генератор сочетаний --";

std::cout << std::endl << "Исходное множество:";

std::cout << "{ ";

for (int i = 0; i < sizeof(AA) / 2; i++)

std::cout << AA[i] << ((i < sizeof(AA) / 2 - 1) ? ", " : " ");

std::cout << "}";

std::cout << std::endl << "Генерация сочетаний:";

combi2::xcombination xc(sizeof(AA) / 2, 3);

std::cout << "из " << xc.n << " по " << xc.m;

int n = xc.getfirst();

while (n >= 0)

{

std::cout << std::endl << xc.nc << ": { ";

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

std::cout << AA[xc.ntx(i)] << ((i < n - 1) ? ", " : " ");

std::cout << "}";

n = xc.getnext();

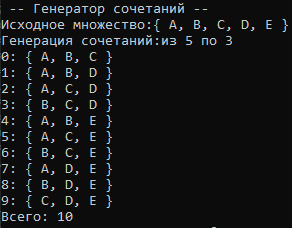
};

std::cout << std::endl << "Всего: " << xc.count() << std::endl;

system("pause");

return 0;

}

****

**Задание 3**. Разработать генератор перестановок,

файл **LB2\_3.cpp**, содержащий вызов функций:

#include <iostream>

#include "Combi3.h"

#include <iomanip>

#include <tchar.h>

int \_tmain(int argc, \_TCHAR\* argv[])

{

setlocale(LC\_ALL, "rus");

char AA[][2] = { "A", "B", "C", "D" };

std::cout << std::endl << " -- Генератор перестановок -- ";

std::cout << std::endl << "Исходное множество: ";

std::cout << "{ ";

for (int i = 0; i < sizeof(AA) / 2; i++)

std::cout << AA[i] << ((i < sizeof(AA) / 2 - 1) ? ", " : " ");

std::cout << "}";

std::cout << std::endl << "Генерация перестановок:";

combi3::permutation p(sizeof(AA) / 2);

\_\_int64 n = p.getfirst();

while (n >= 0)

{

std::cout << std::endl << std::setw(4) << p.np << ": { ";

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

std::cout << AA[p.ntx(i)] << ((i < p.n - 1) ? ", " : " ");

std::cout << "}";

n = p.getnext();

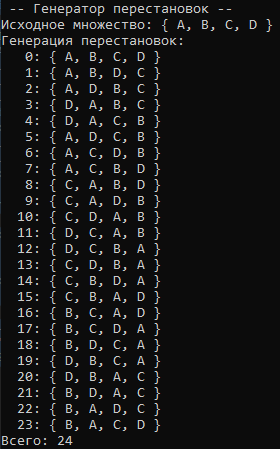
};

std::cout << std::endl << "Всего: " << p.count() << std::endl;

system("pause");

return 0;

}



**Задание 4**. Разработать генератор размещений,

файл **LB2\_4.cpp**, содержащий вызов функций:

#include "stdafx.h"

#include <iostream>

#include <iomanip>

#include <tchar.h>

#include "Combi4.h"

#define N (sizeof(AA)/2)

#define M 3

int \_tmain(int argc, \_TCHAR\* argv[])

{

setlocale(LC\_ALL, "rus");

char AA[][2] = { "A", "B", "C", "D" };

std::cout << " -- Генератор размещений --";

std::cout << std::endl << "Исходное множество: ";

std::cout << "{ ";

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

std::cout << AA[i] << ((i < N - 1) ? ", " : " ");

std::cout << "}";

std::cout << std::endl << "Генерация размещений из " << N << " по " << M;

combi4::accomodation s(N, M);

int n = s.getfirst();

while (n >= 0) {

std::cout << std::endl << std::setw(2) << s.na << ": { ";

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

std::cout << AA[s.ntx(i)] << ((i < n - 1) ? ", " : " ");

std::cout << "}";

n = s.getnext();

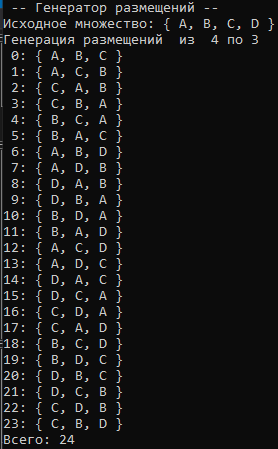
};

std::cout << std::endl << "Всего: " << s.count() << std::endl;

system("pause");

return 0;

}



**Задание 5**. Решить в соответствии с вариантом задачу об оптимальной загрузке судна (веса контейнеров сгенерировать случайным образом: ограничение по общему весу – 1500 кг., количество мест на судне для контейнеров – 5, количество контейнеров 25, веса контейнеров 100 – 900 кг., доход от перевозки 10 – 150 у.е.),

файл **LB2\_5.cpp**, содержащий вызов функций:

#include <iostream>

#include <iomanip>

#include <tchar.h>

#include "Boat.h"

#define NN 25

#define MM 5

int \_tmain(int argc, \_TCHAR\* argv[])

{

setlocale(LC\_ALL, "rus");

srand(time(NULL));

int V = 1500;

// v[NN] = { 100, 200, 300, 400, 500, 150 },

// c[NN] = { 10, 15, 20, 25, 30, 25 };

int v[NN], c[NN];

for (int i = 0; i < NN; i++) {

v[i] = rand() % 800 + 100;

c[i] = rand() % 140 + 10;

}

short r[MM];

int cc = boat(

V, // [in] максимальный вес груза

MM, // [in] количество мест для контейнеров

NN, // [in] всего контейнеров

v, // [in] вес каждого контейнера

c, // [in] доход от перевозки каждого контейнера

r // [out] результат: индексы выбранных контейнеров

);

std::cout << std::endl << "- Задача о размещении контейнеров на судне";

std::cout << std::endl << "- общее количество контейнеров : " << NN;

std::cout << std::endl << "- количество мест для контейнеров : " << MM;

std::cout << std::endl << "- ограничение по суммарному весу : " << V;

std::cout << std::endl << "- вес контейнеров : ";

for (int i = 0; i < NN; i++) std::cout << std::setw(3) << v[i] << " ";

std::cout << std::endl << "- доход от перевозки : ";

for (int i = 0; i < NN; i++) std::cout << std::setw(3) << c[i] << " ";

std::cout << std::endl << "- выбраны контейнеры (0,1,...,m-1): ";

for (int i = 0; i < MM; i++) std::cout << r[i] << " ";

std::cout << std::endl << "- доход от перевозки : " << cc;

std::cout << std::endl << "- общий вес выбранных контейнеров : ";

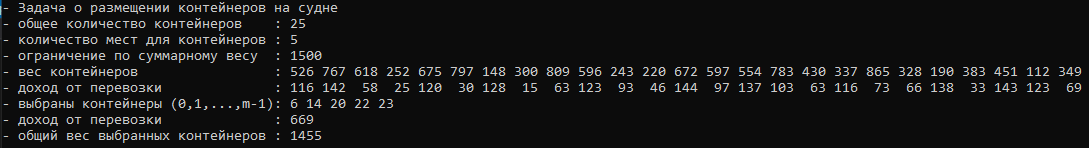
int s = 0; for (int i = 0; i < MM; i++) s += v[r[i]]; std::cout << s;

std::cout << std::endl << std::endl;

system("pause");

return 0;

}



**Задание 6**. Исследовать зависимость времени вычисления необходимое для решения задачи (в соответствии с вариантом) от размерности задачи об оптимальной загрузке судна (количество мест на судне для контейнеров – 6, количество контейнеров 25 – 35,

файл **LB2\_5.cpp**, содержащий вызов функций:

#include <iostream>

#include <iomanip>

#include "Boat.h"

#include "tchar.h"

#include <time.h>

#define NN 36

#define MM 6

#define SPACE(n) std::setw(n) << " "

int \_tmain(int argc, \_TCHAR\* argv[])

{

setlocale(LC\_ALL, "rus");

srand(time(NULL));

int V = 1500;

//v[NN] = { 250, 560, 670, 400, 200, 270, 370, 330, 330, 440, 530, 120,

// 200, 270, 370, 330, 330, 440, 700, 120, 550, 540, 420, 170,

// 600, 700, 120, 550, 540, 420, 430, 140, 300, 370, 310, 120 },

//c[NN] = { 15,26, 27, 43, 16, 26, 42, 22, 34, 12, 33, 30,

// 42,22, 34, 43, 16, 26, 14, 12, 25, 41, 17, 28,

// 12,45, 60, 41, 33, 11, 14, 12, 25, 41, 30, 40 };

int v[NN], c[NN];

for (int i = 0; i < NN; i++) {

v[i] = rand() % 800 + 100;

c[i] = rand() % 140 + 10;

}

short r[MM];

int maxcc = 0;

clock\_t t1, t2;

std::cout << std::endl << "-- Задача об оптимальной загрузке судна -- ";

std::cout << std::endl << "- ограничение по весу : " << V;

std::cout << std::endl << "- количество мест : " << MM;

std::cout << std::endl << "-- количество ------ продолжительность -- ";

std::cout << std::endl << " контейнеров вычисления ";

for (int i = 25; i <= NN; i++)

{

t1 = clock();

int maxcc = boat(V, MM, i, v, c, r);

t2 = clock();

std::cout << std::endl << SPACE(7) << std::setw(2) << i << SPACE(15) << std::setw(5) << (t2 - t1);

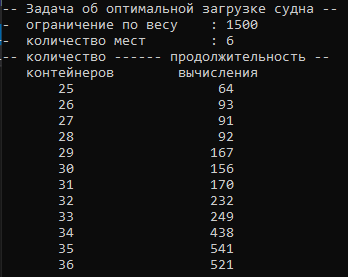
}

std::cout << std::endl << std::endl;

system("pause");

return 0;

}



****

**График** зависимости продолжительности процесса вычисления от количества контейнеров

**Вывод**: при выполнении данной лабораторной работы я приобрела навыки разработки генераторов подмножеств, перестановок, сочетаний и размещений на С++ и научилась применять разработанные генераторы для решения задачи о рюкзаке об оптимальной загрузке судна.