Учреждение образования

«Белорусский государственный технологический университет»

**Лабораторная работа №1**

Вспомогательные функции

Выполнил:

Студент 2 курса 4 группы ФИТ

Кивлинас Олег Леонидович

2024 г.

**Задание 1**

Разработайте три функции (start, dget и iget), используя следующие спецификации (рисунок 1):

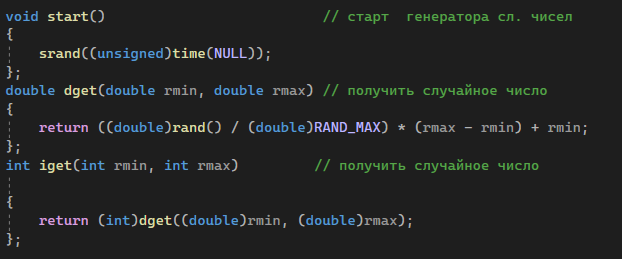


Рисунок 1 – Реализация функций

**Задание 2**

1.Реализовать пример 2.

2.Для проверки работоспособности разработанных функций и приобретения навыков замера продолжительности процесса вычисления реализуйте программу, приведенную в примере 2.

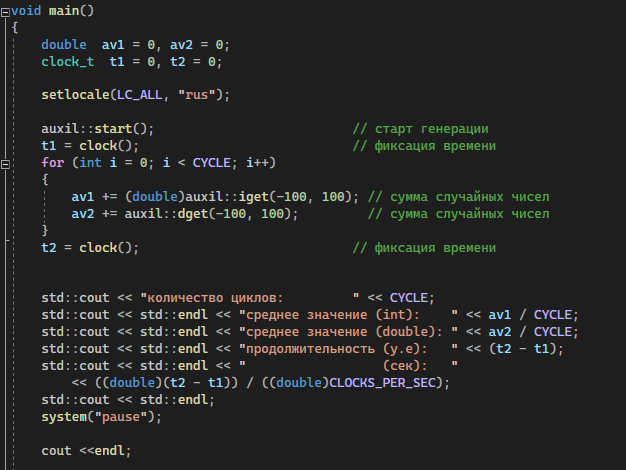


Рисунок 2 – Вызов функции и измерение времени выполнения

**Задание 3**

Проведите необходимые эксперименты (разработать кодом) и постройте график зависимости (Excel) продолжительности процесса вычисления от количества циклов в примере 2.

Примечание: продолжительность вычисления измерять в условных единицах процессорного времени (функция steady\_clock). График представлен на рисунке 3

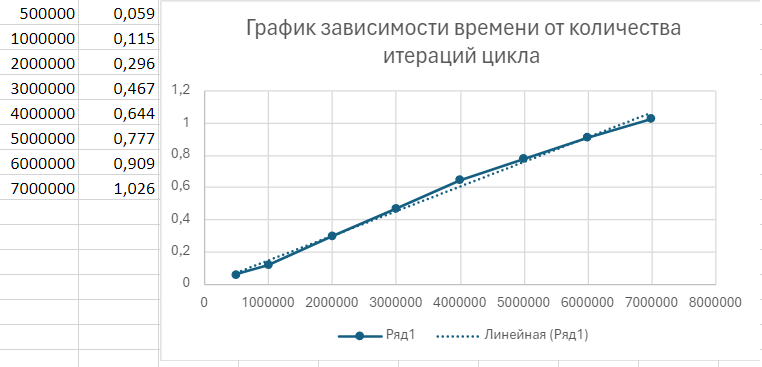


Рисунок 3 – график зависимости

Реализация Факториала в коде демонстрируется на рис. 4

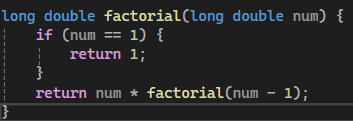


Рисунок 4 – Функция для нахождения факториала

Вызов и подсчет времени для реализации факториала представлен на рисунке 5:

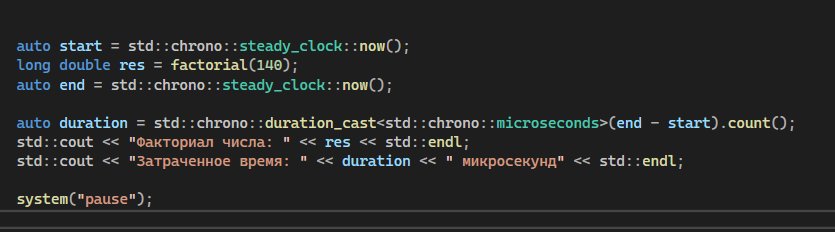


Рисунок 5 – Вызов функции для подсчёта факториала

График зависимости:

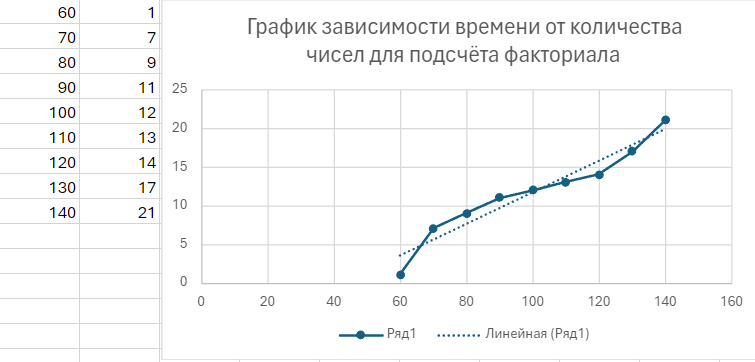


Рисунок 6 – График зависимости

**Вывод**: по полученным измерениям, можем сделать следующий вывод, что время выполнения программы линейно зависит от количества циклов или значения факториала.

# Лабораторная работа 2

**Комбинаторные алгоритмы решения оптимизационных задач**

**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. Генератор подмножеств заданного множества будет на листинге 2.1.

|  |
| --- |
| #include <iostream>  #include "Comb.h"  #include <tchar.h>  using std::cout;  using std::endl;  int \_tmain(int argc, \_TCHAR\* argv[])  {  setlocale(LC\_ALL, "rus");  char AA[][2] = {"A", "B", "C", "D", "E"};  cout << "Генератор множества всех подмножеств\n";  cout << endl << "Исходное множество: ";  cout << "{ ";  for (int i = 0; i < sizeof(AA) / 2; i++)  cout << AA[i] << ((i < sizeof(AA) / 2 - 1) ? ", " : " ");  cout << "}\n\nИтоговая генерация всех подмножеств исходного множества:";  comb::subset s1(sizeof(AA) / 2); // создание генератора  int n = s1.getfirst(); // первое (пустое) подмножество  while (n >= 0) // пока есть подмножества  {  cout << endl << "{ ";  for (int i = 0; i < n; i++)  cout << AA[s1.ntx(i)] << ((i < n - 1) ? ", " : " ");  cout << "}";  n = s1.getnext(); // cледующее подмножество  };  cout << endl << "Всего множеств: " << s1.count() << endl;  system("pause");  return 0;  } |

Листинг 2.1 – Генератор подмножеств

Результат выполнения генерации представлен на рисунке 2.1.



Рисунок 2.1 – Генератор подмножеств заданного множества

Алгоритм представлен на рисунке 2.2.



Рисунок 2.2 – Генератор подмножеств заданного множества

**Задание 2:** Разработать генератор сочетаний. Код будет представлен на листинге 2.2.

|  |
| --- |
| int \_tmain(int argc, \_TCHAR\* argv[])  {  setlocale(LC\_ALL, "rus");  char AA[][2] = { "A", "B", "C", "D", "E"};  cout << "Генератор сочетаний\n";  cout << endl << "Исходное множество: ";  cout << "{ ";  for (int i = 0; i < sizeof(AA) / 2; i++)  cout << AA[i] << ((i < sizeof(AA) / 2 - 1) ? ", " : " ");  cout << "}\n";  cout << endl << "Генерация сочетаний ";  comb::xcombination xc(sizeof(AA) / 2, 3);  cout << "из " << xc.n << " по " << xc.m;  int n = xc.getfirst();  while (n >= 0)  {  cout << endl << xc.nc << ": { ";  for (int i = 0; i < n; i++)  cout << AA[xc.ntx(i)] << ((i < n - 1) ? ", " : " ");  cout << "}";  n = xc.getnext();  };  cout << endl << "Всего сочетаний: " << xc.count() << endl;  system("pause");  return 0;  } |

Листинг 2.2 – Код генератора сочетаний

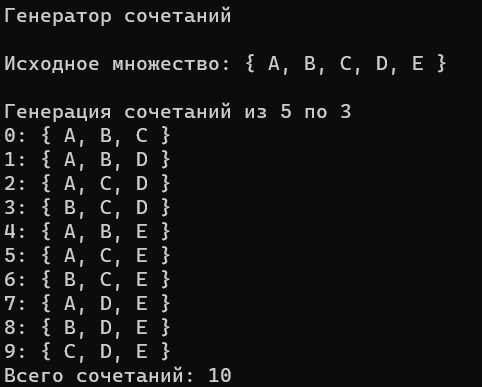


Рисунок 2.3 –Результат генератора сочетаний

Алгоритм будет представлен на рисунке 2.4

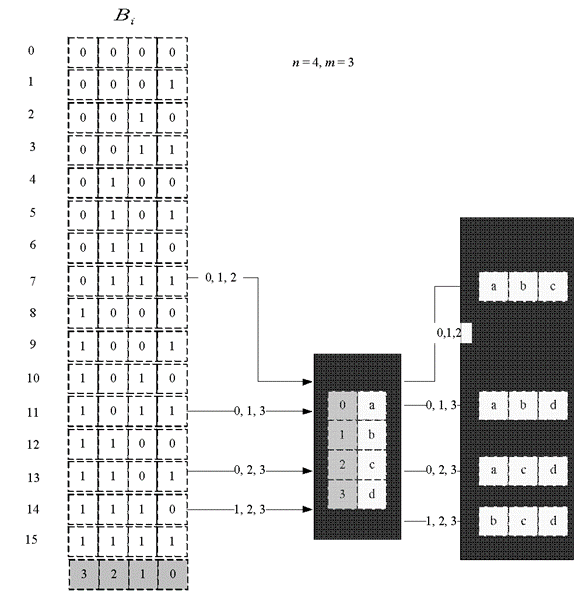


Рисунок 2.4 – Алгоритм генератора сочетаний

**Задание 3:** Разработать генератор перестановок. Код будет представлен на листинге 2.3.

|  |
| --- |
| int \_tmain(int argc, \_TCHAR\* argv[])  {  setlocale(LC\_ALL, "rus");  char AA[][2] = { "A", "B", "C", "D" };  cout << "Генератор перестановок\n";  cout << endl << "Исходное множество: {";  for (int i = 0; i < sizeof(AA) / 2; i++)  cout << AA[i] << ((i < sizeof(AA) / 2 - 1) ? ", " : " ");  cout << "}";  cout << endl << "\nГенерация перестановок ";  comb::permutation p(sizeof(AA) / 2);  \_\_int64 n = p.getfirst();  while (n >= 0)  {  cout << endl << setw(4) << p.np << ": { ";  for (int i = 0; i < p.n; i++)  cout << AA[p.ntx(i)] << ((i < p.n - 1) ? ", " : " ");  cout << "}";  n = p.getnext();  };  cout << endl << "Всего перестановок: " << p.count() << endl;  system("pause");  return 0;} |

Листинг 2.3 – Код генератора перестановок

Результат выполнения представлен на рисунке 2.5.

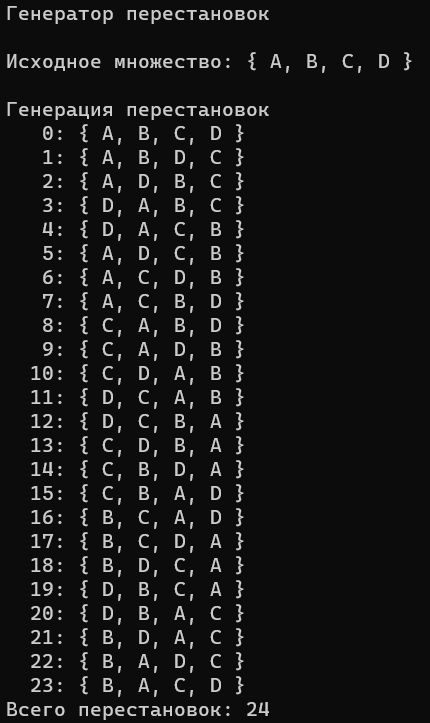


Рисунок 2.5 –Результат генератора перестановок



Рисунок 2.9 – Генератор перестановок

**Задание 4:** Разработать генератор размещений. Код будет представлен на рисунке 2.10.

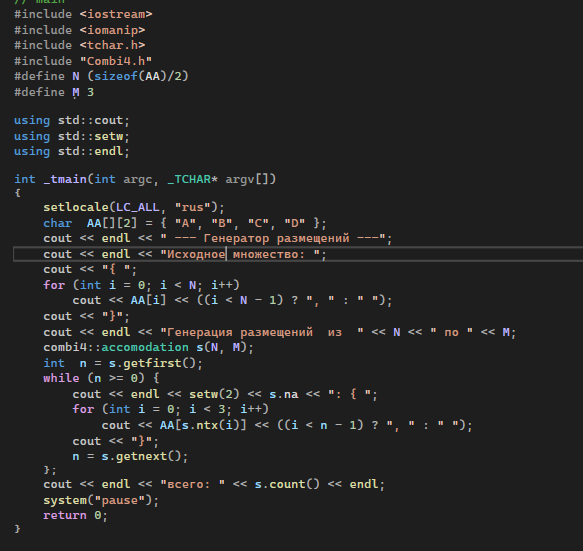


Рисунок 2.10 – Код генератора размещений

Результат выполнения будет представлен на рисунке 2.11.

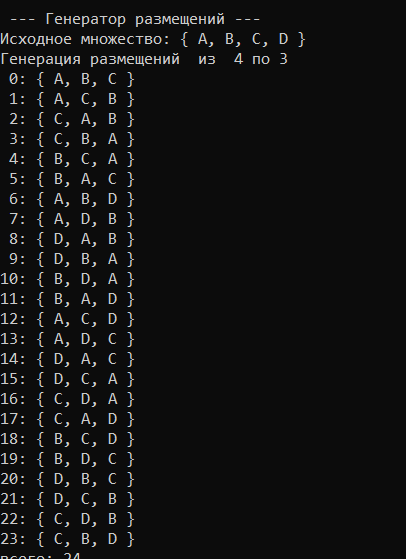


Рисунок 2.11 –Результат генератора размещений

Алгоритм будет представлен на рисунке 2.12.



Рисунок 2.12– Генератор размещений

**Задание 5:** (2, 6, 10, 14) упрощенную о рюкзаке (веса предметов и их стоимость сгенерировать случайным образом: вместимость рюкзака 300 кг, веса предметов 10 – 300 кг, стоимость предметов 5 – 55 у.е.; количество предметов – 18 шт.). Код этого задания будет на рисунке 2.13.

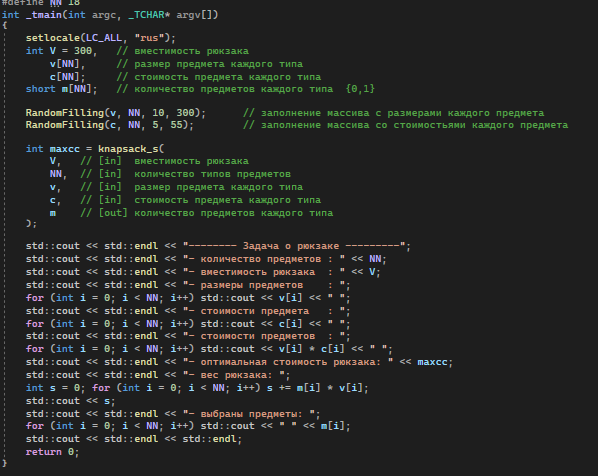


Рисунок 2.13 – Код задания 5

Результат выполнения будет представлен на рисунке 2.14.

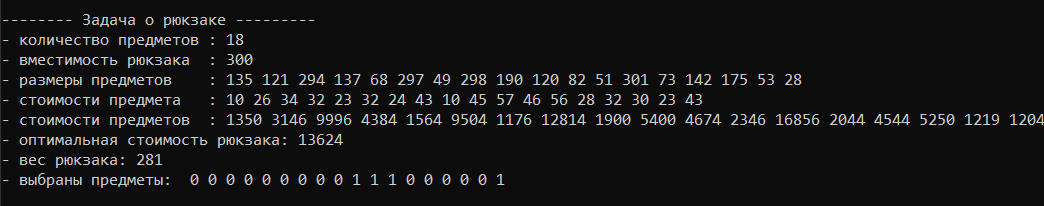


Рисунок 2.14 –Результат задания 5

**Задание 6: (**2, 6, 10, 14) упрощенную о рюкзаке (количество предметов 12 – 20 шт.). Код задания номер 6 будет представлен на рисунке 2.15.

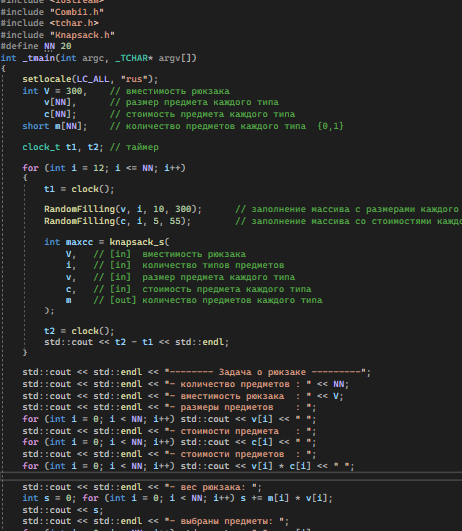


Рисунок 2.15 – Код задания 6

Результат выполнения будет представлен на рисунке 2.16.

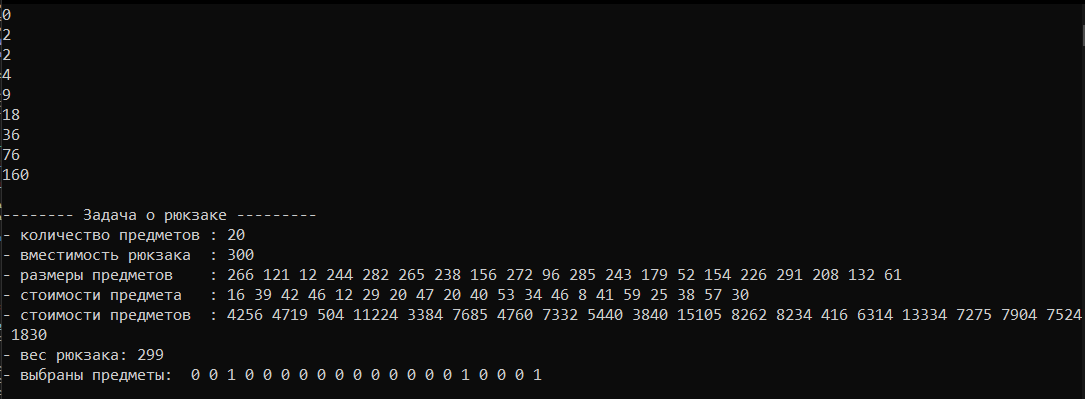


Рисунок 2.16–Результат задания 6

График будет представлен на рисунке 2.17.

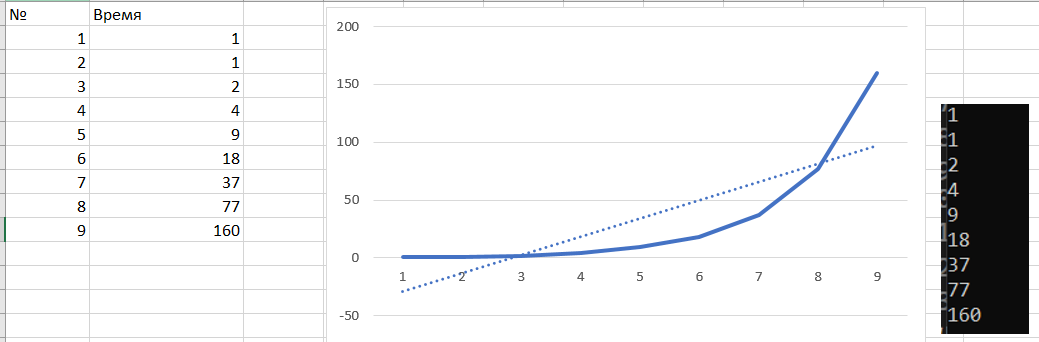


Рисунок 2.17 – График

Алгоритм будет представлен на рисунке 2.18.



Рисунок 2.18– Схема решения задачи о рюкзаке

**Вывод:** приобретены навыки разработки генераторов подмножеств, перестановок, сочетаний и размещений на С++; применены разработанные генераторы об оптимальной загрузке судна, построена зависимость.