КПІ ім. Ігоря Сікорського

Інститут прикладного системного аналізу

Кафедра Системного проектування

Алгоритми та структури даних

**Лабораторна робота №1**

**«Дослідження базових алгоритмів»**

Виконав:

Студент групи ДА-02

Рудік Андрій

Варіант №16

Мета роботи:

Ознайомитись і дослідити базові класичні алгоритми: пошук простих

чисел, переведення числа з однієї системи числення в іншу, рекурсивні

алгоритми. Набути навичок їх реалізації мовою програмування С/C++,

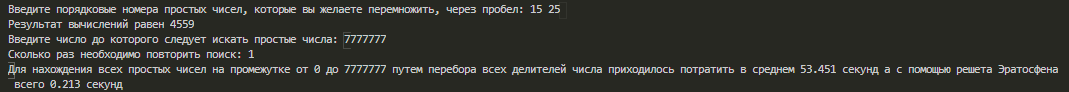
навчитися вимірювати та порівнювати швидкодію алгоритмів

Варіант №16.

Завдання №1

1. Знайти добуток 15-го та 25-го за рахунком простих чисел.
2. Реалізувати булеву функцію bool isPrime(int n), яка перевіряє вхідне число n на простоту за допомогою звичайного перебору дільників.
3. Реалізувати алгоритм “Решето Ератосфена” для знаходження всіх простих чисел на проміжку від 1 до N.
4. Розв’язати задану задачу у 2 способи: за допомогою функції isPrime() та за допомогою решета Ератосфена, заміряти та вивести час виконання кожного рішення.

Результат



Лістинг

#include <iostream>

#include <ctime>

#include <cmath>

#include <clocale>

using *namespace* std;

*bool* isPrime(*int* *n*) {

  if ((*n* % 2 == 0 && *n* != 2) || *n* == 1) return false;

  for (*int* d = 3; d <= sqrt(*n*); d += 2) {

    if (*n* % d == 0) {

      return false;

    }

  }

  return true;

}

*int*\* getPrimesDumb(*int* *N*) {

*int* \*primes = new *int*[*N*];

  for (*int* i = 2, index = 0; i < *N*; i++) {

    if (isPrime(i)) {

      primes[index] = i;

      index++;

    }

  }

  return primes;

}

*int*\* sieveOfEratosthenes(*int* *N*) {

*bool* \*sieve = new *bool*[*N*];

  sieve[0] = false;

  sieve[1] = false;

  for (*int* i = 2; i < *N*; i++) {

    sieve[i] = true;

  }

  for (*int* i = 2; i <= sqrt(*N*); i++) {

    if (!sieve[i]) continue;

    for (*int* j = i \* i; j < *N*; j += i) {

      sieve[j] = false;

    }

  }

*int* \*primes = new *int*[*N*];

  for (*int* i = 0, index = 0; i < *N*; i++) {

    if (!sieve[i]) continue;

    primes[index] = i;

    index++;

  }

  delete [] sieve;

  return primes;

}

// n - number of primes

*int* getRange(*int* *n*) {

*int* primesCounter = 0;

*int* i = 0;

  while (primesCounter < *n*) {

    if (isPrime(i)) primesCounter++;

    i++;

  }

  return i;

}

*int* main() {

  setlocale(LC\_ALL, "rus");

*int* a;

*int* b;

  cout << "Введите порядковые номера простых чисел, которые вы желаете перемножить, через пробел: ";

  cin >> a >> b;

*int* \*primes = a > b ? sieveOfEratosthenes(getRange(a)) : sieveOfEratosthenes(getRange(b));

  cout << "Результат вычислений равен " << primes[a - 1] \* primes[b - 1] << endl;

*int* N = 0;

  cout << "Введите число до которого следует искать простые числа: ";

  cin >> N;

*int* numberOfTests = 1;

  cout << "Сколько раз необходимо повторить поиск: ";

  cin >> numberOfTests;

  clock\_t start = clock();

  for (*int* i = 0; i < numberOfTests; i++) {

    getPrimesDumb(N);

  }

  clock\_t end = clock();

*float* dumbAverage = ((*float*)(end - start)) / numberOfTests / CLOCKS\_PER\_SEC;

  start = clock();

  for (*int* i = 0; i < numberOfTests; i++) {

    sieveOfEratosthenes(N);

  }

  end = clock();

*float* eratosthenesAverage = ((*float*)(end - start)) / numberOfTests / CLOCKS\_PER\_SEC;

  cout << "Для нахождения всех простых чисел на промежутке от 0 до " << N <<

    " путем перебора всех делителей числа приходилось потратить в среднем " << dumbAverage << " секунд " <<

    "а с помощью решета Эратосфена всего " << eratosthenesAverage << " секунд";

  return 0;

}

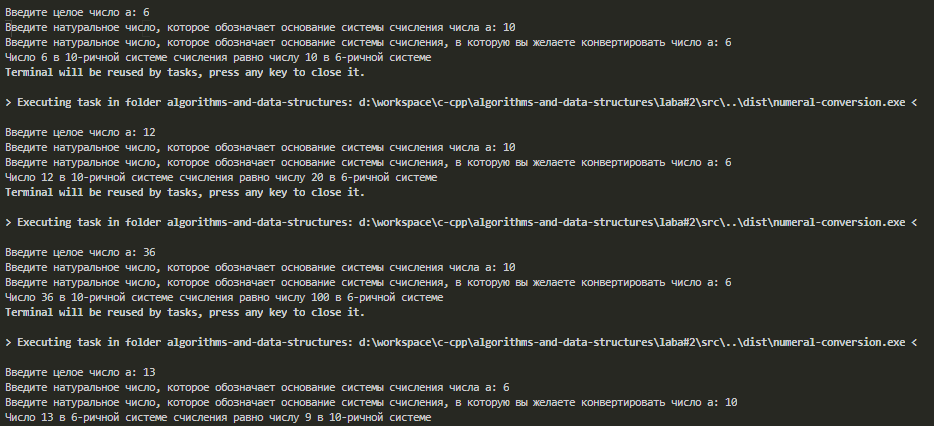
Завдання №2

Написати функцію для переведення натурального десяткового числа в

6-ву систему числення, і функцію для зворотнього переводу.

Протестувати на декількох прикладах.

Результат



Лістинг

#include <iostream>

#include <clocale>

#include <cmath>

#include <string>

using *namespace* std;

const *int* NUMS\_ARRAY\_LENGTH = 36;

const *char* NUMS\_ARRAY[36] = {

  '0', '1', '2', '3', '4', '5', '6', '7', '8',

  '9', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h',

  'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q',

  'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z',

};

*int* getValueOfNumeral(*char* *num*) {

  for (*int* i = 0; i < NUMS\_ARRAY\_LENGTH; i++) {

    if (NUMS\_ARRAY[i] == num) {

      return i;

    }

  }

  // Вот тут по-хорошему ошибочку бы выбросить...

  return 0;

}

string reverse(string *str*) {

  string reversed = "";

  for (*int* i = str.length() - 1; i >= 0; i--) {

    reversed += str[i];

  }

  return reversed;

}

string convertBase(string *a*, *unsigned* *short* *int* *fromBase*, *unsigned* *short* *int* *toBase*) {

  if (fromBase == toBase) {

    return a;

  }

  // Конвертируем в десятичную

  a = reverse(a);

*unsigned* *int* aDecimial = 0;

  for (*int* i = 0; i < a.length(); i++) {

    aDecimial += getValueOfNumeral(a[i]) \* pow(fromBase, i);

  }

  string converted = "";

  while (aDecimial) {

    converted += NUMS\_ARRAY[aDecimial % toBase];

    aDecimial /= toBase;

  }

  return reverse(converted);

}

*int* main() {

  setlocale(LC\_ALL, "rus");

  string a;

*unsigned* *short* *int* fromBase;

*unsigned* *short* *int* toBase;

  cout << "Введите целое число а: ";

  cin >> a;

  cout << "Введите натуральное число, которое обозначает основание системы счисления числа a: ";

  cin >> fromBase;

  cout << "Введите натуральное число, которое обозначает основание системы счисления, в которую вы желаете конвертировать число a: ";

  cin >> toBase;

  if (fromBase == 0 || toBase == 0 || fromBase > NUMS\_ARRAY\_LENGTH || toBase > NUMS\_ARRAY\_LENGTH) {

    cout << "Некоректное основание системы счисления";

    return 1;

  }

  cout << "Число " << a << " в " << fromBase << "-ричной системе счисления равно числу " <<

    convertBase(a, fromBase, toBase) << " в " << toBase << "-ричной системе";

  return 0;

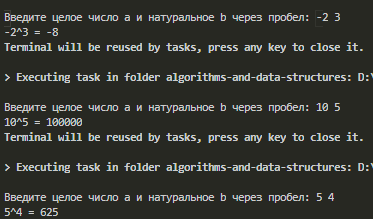
}

Завдання №3

Дано невід’ємні цілі числа A та B. Обчислити A в степені B за допомогою

рекурсії (без використання циклів).

Результат



Лістинг

#include <iostream>

#include <clocale>

using *namespace* std;

// Принимает целое число а и натуральное b

*int* recursivePow(*int* *a*, *unsigned* *int* *b*) {

  if (*b* == 0) {

    return 1;

  }

  return *b* == 1 ? *a* : recursivePow(*a*, *b* - 1) \* *a*;

}

*int* main() {

  setlocale(LC\_ALL, "rus");

*int* a;

*unsigned* *int* b;

  cout << "Введите целое число a и натуральное b через пробел: ";

  cin >> a >> b;

  cout << a << "^" << b << " = " << recursivePow(a, b);

  return 0;

}

Висновок

Я ознайомився і дослідив базові класичні алгоритми: пошук простих

чисел, переведення числа з однієї системи числення в іншу, рекурсивні

алгоритми. Набув навичок їх реалізації мовою програмування С/C++,

навчився вимірювати та порівнювати швидкодію алгоритмів.