**Цель работы:** требуется проанализировать быстродействие решения задачи Древний шифр (методов шифрования является атбаш).

**Техническое задание:** Написать программу дешифрования шифра атбаш на языке с++ в Vusual Studio. Шифр реализуется достаточно прост: первая буква алфавита («a») в нём заменяется на последнюю («z»), вторая («b») — на предпоследнюю («y»), ..., последняя («z») — на первую («a»). Формально, буква, находящаяся на позиции X от начала алфавита, заменяется буквой, находящейся на позиции X от конца алфавита. Символы, отличные от букв, не изменяются.

1) **Код тестирования:**

#include<stdio.h>

#include <iostream>

#include <ctime>

#include <string>

using namespace std;

const int ALL = 25;

string generateStr(int size) {

srand(time(NULL));

int number;

string str;

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

number = rand() % 7;

if (number == rand() % 7) {

str += ' ';

}

else {

str += 'a' + rand() % 26;

}

}

str[size] = '\0';

return str;

}

void check(string str) {

for (int i = 0; str[i]; i++) {

if (isupper(str[i])) {

if (!isalpha(str[i])) {

continue;

}

str[i] -= 'A';

str[i] = (ALL - str[i]) % 26;

str[i] += 'A';

}

else {

if (!isalpha(str[i])) {

continue;

}

str[i] -= 'a';

str[i] = (ALL - str[i]) % 26;

str[i] += 'a';

}

}

}

#include <chrono>

using namespace std::chrono;

double singleTimeTest(int size) {

string stroka = generateStr(size);

high\_resolution\_clock::time\_point startTime = high\_resolution\_clock::now();

check(stroka);

high\_resolution\_clock::time\_point finishTime = high\_resolution\_clock::now();

duration<double> elapsedTime = duration\_cast<duration<double>>(finishTime - startTime);

return elapsedTime.count();

}

double multipleTimeTests(int size, int testsCount) {

double totalTime = 0;

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

totalTime += singleTimeTest(size);

return totalTime / testsCount;

}

void timeTestingTable() {

printf("N\tT(N)\n");

for (int n = 16; n <= 8192; n \*= 2)

printf("%d\t%lf\n", n, multipleTimeTests(n, 100));

}

int main() {

timeTestingTable();

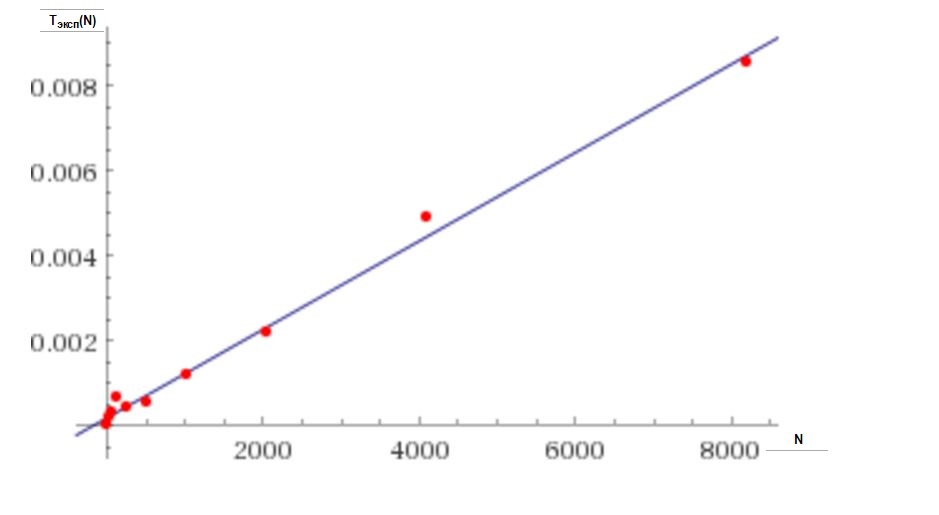
return 0;

}

**Результат работы программы:**

|  |  |
| --- | --- |
| N | Tэксп(N) |
| 16 | 0.000030 |
| 32 | 0.000171 |
| 64 | 0.000276 |
| 128 | 0.000669 |
| 256 | 0.000412 |
| 512 | 0.000521 |
| 1024 | 0.001156 |
| 2048 | 0.002173 |
| 4096 | 0.004878 |
| 8192 | 0.008553 |

**2) График:**



3) **Таблица погрешностей:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Tэксп(N) | Tтеор(N) | ∆ | δ |
| 16 | 0,000030 | 0,000194 | 0,000164 | 5,466667 |
| 32 | 0,000171 | 0,000211 | 0,000040 | 0,233918 |
| 64 | 0,000276 | 0,000244 | 0,000032 | 0,115942 |
| 128 | 0,000669 | 0,000311 | 0,000358 | 0,535127 |
| 256 | 0,000412 | 0,000444 | 0,000032 | 0,07767 |
| 512 | 0,000521 | 0,000711 | 0,000190 | 0,364683 |
| 1024 | 0,001156 | 0,001245 | 0,000089 | 0,07699 |
| 2048 | 0,002173 | 0,002312 | 0,000139 | 0,063967 |
| 4096 | 0,004878 | 0,002312 | 0,002566 | 0,526035 |
| 8192 | 0,008553 | 0,008715 | 0,000162 | 0,018941 |

Вывод: на основании проведённых экспериментов можно сделать вывод о том, что зависимость времени работы программы T от размера входных данных N описывается функцией y = 0.000178193 + 1.0421×10^-6 x.