

Отчет по лабораторной работе № 1

дисциплина:

Наливайко Сергей Максимович

Содержание

1	Цель работы	3
2	Задание	4
3	Выполнение лабораторной работы	5
3.1	Шифр Цезаря	5
3.2	Шифр Атбаш	5
3.3	Запуск программы	6
4	Выводы	8

1 Цель работы

Научиться реализовывать простейшие алгоритмы шифрования.

2 Задание

- Реализовать шифр Цезаря с произвольным ключом k .
- Реализовать шифр Атбаш.

3 Выполнение лабораторной работы

3.1 Шифр Цезаря

Реализуем алгоритм шифрования с помощью языка программирования C++. Код для функции шифрования представлен ниже.

```
std::string Caesar_Cipher::crypt(const std::string &text, int k) {  
    std::stringstream str;  
    int index;  
    k %= ALPHABET_SIZE;  
    for(char i : text) {  
        index = (i - ' ' + k) % ALPHABET_SIZE + ' '  
        str << static_cast<char>(index);  
    }  
    return str.str();  
}
```

Алфавит имеет длину 95 символов. Доступ к алфавиту реализован через таблицу ASCII.

3.2 Шифр Атбаш

Реализуем алгоритм шифрования с помощью языка программирования C++. Код для функции шифрования представлен ниже.

```

std::string Atbash_Cipher::crypt(const std::string &text) {
    std::stringstream str;
    int index;
    for(char i : text) {
        index = ALPHABET_SIZE - (i - ' ') - 1 + ' ';
        str << static_cast<char>(index);
    }
    return str.str();
}

```

Алфавит имеет длину 95 символов. Доступ к алфавиту реализован через таблицу ASCII.

3.3 Запуск программы

Полная реализация программы находится в прикрепленном архиве. Здесь будет отображен лишь запуск.

Введем строку в программу: hello, WORLD! My name is Sergey!~P{[[].

Полученные шифры для двух случаев отображены на fig. 3.1.

```

sergey@sergey: ~/University/MathSec/lab01/program/cmake-build-debug
File Edit View Search Terminal Help
sergey@sergey:~/University/MathSec/lab01/program/cmake-build-debug$ ./program
hello, WORLD! My name is Sergey!~P{[]}
Your input: hello, WORLD! My name is Sergey!~P{[]}
Caesar cipher for:
Shift = 0: hello, WORLD! My name is Sergey!~P{[]}
Shift = 1: ifmmp-!XPSME"!Nz!obnf!jt!Tfshfz" Q|~\^
Shift = 2: jgnnq,"YQTNF#"0{"pcog"ku"Ugtig{#!R} ]
Shift = 3: kloor/#ZRUOG$#P|#qdp#lv#Vhujh|$~S-!^~
Shift = 4: lipps0$[SVPH$Q}$reqi$mw$Wivki}$#T " _a
Shift = 5: mjqq1%\TWQI&%R~%sfrj%nx%XjwLj~&$U!#~b
Shift = 6: nkrru2&|UXRJ'&S &tgsK&oy&Ykxmk '%V"$sac
Shift = 7: olssv3^VYSK('T!'uhtl'pz'Zlyn!(&W#%bd
Shift = 8: pmttw4(_WZTL)(U"(vium(q{([mzom")'X&ce
Shift = 9: qnuux5)`X[UM*)V#)w|vn)r|)\n{pn#*(Y%'df
Shift = 10: rovv6*aY\VN+*W$*xkwo*s)*|o|qo$+)Z&(eg
Shift = 11: spwwz7+bZ]W0,+X%+yLxp+t-+^p)rp%,*[')fh
Shift = 12: tqxx{8,c[^XP-,Y&,zmyq,u ,_q~sq&-+(*gi
Shift = 13: uryy|9-d\Y0,-Z'-(nzt-r tr'.,))+hj
Shift = 14: vszz|:.e)`ZR/.{(.|o{s.w".as!us(/-^*,ik
Shift = 15: wt{{~;/f^a[S0/)/}p|t/x#/bt"vt)0. _+~jL
Shift = 16: xu|| <0g_bTl0]*0-q|u0y$0cu#wu+1/~, .km
Shift = 17: yv}}!~1h`c]U21^+1 r~v1z%1dv$ xv+20a-/ln
Shift = 18: zw~">2iad^V32 ,2!s w2{&2ew$yw,31b.0mo
Shift = 19: {x #73jbe W43^-3"t!x3|'3fx&z-x-42c/lnp
Shift = 20: |y!!$@4kcf`X54a.4#u"y4}(4gy'{y.53d02oq
Shift = 21: }z""%A5ldgaY65b/5$#z5~)5hz(|z/64e13pr
Shift = 22: ~{#&B6mehbZ76c06%w$[6 *6i{)}{075f24qs
Shift = 23: |$'$C7nfic[87d17&x%|7!+7j|*~|186g35rt
Shift = 24: !}%%(D8ogjd\98e28'y&}8",8k)+ }297h46su
Atbash cipher: 6922/r~GOLRZ]-0~0=19~5+~K9,79%} N#!CA
sergey@sergey:~/University/MathSec/lab01/program/cmake-build-debug$

```

Figure 3.1: Запуск программы

4 Выводы

В ходе лабораторной работы мы научились реализовывать простейшие алгоритмы шифрования.