BVRIT Hyderabad

BH23 PPS Project

Caesar Cipher Project

Team:

Veda Sathvika M - 23WH1A6634

A Jyotsna Sravani – 23WH1A6657

Sai Srinidhi – 23WH1A6631

T Ashwitha Priya – 23WH1A6629

Problem Statement :

**Project 10: Caesar Cipher in Cryptography**

The Caesar cipher is a simple encryption technique that was used by Julius Caesar to send secret messages to his allies. It works by shifting the letters in the plaintext message by a certain number of positions, known as the “shift” or “key”.

The Caesar Cipher technique is one of the earliest and simplest methods of encryption technique. It’s simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter with a fixed number of positions down the alphabet. For example with a shift of 1, A would be replaced by B, B would become C, and so on. The method is apparently named after Julius Caesar, who apparently used it to communicate with his officials.

Thus to cipher a given text we need an integer value, known as a shift which indicates the number of positions each letter of the text has been moved down.

The encryption can be represented using modular arithmetic by first transforming the letters into numbers, according to the scheme, A = 0, B = 1,…, Z = 25. Encryption of a letter by a shift n can be described mathematically as.

For example, if the shift is 3, then the letter A would be replaced by the letter D, B would become E, C would become F, and so on. The alphabet is wrapped around so that after Z, it starts back at A.

Here is an example of how to use the Caesar cipher to encrypt the message “HELLO” with a shift of 3:

Write down the plaintext message: HELLO

Choose a shift value. In this case, we will use a shift of 3.

Replace each letter in the plaintext message with the letter that is three positions to the right in the alphabet.

         H becomes K (shift 3 from H)

         E becomes H (shift 3 from E)

         L becomes O (shift 3 from L)

         L becomes O (shift 3 from L)

O becomes R (shift 3 from O)

      4.The encrypted message is now “KHOOR”.

To decrypt the message, you simply need to shift each letter back by the same number of positions. In this case, you would shift each letter in “KHOOR” back by 3 positions to get the original message, “HELLO”.

Text : ABCDEFGHIJKLMNOPQRSTUVWXYZ

Shift: 23

Cipher: XYZABCDEFGHIJKLMNOPQRSTUVW

Text : ATTACKATONCE

Shift: 4

Cipher: EXXEGOEXSRGI

Source Code:

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

void caesar\_cipher(char\* ,int);

int main()

{

int shift,n,i;

char \*text;

printf("Enter the shift: ");

scanf("%d",&shift);

printf("Enter n:");

scanf("%d",&n);

text = (char\*) calloc(n,sizeof(char));

printf("Enter the text: ");

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

{

scanf("%c",(text+i));

}

n=(sizeof(text))/2;

int \*ptr;

ptr = (int\*) calloc(n,sizeof(int));

for(int j=1;j<=n;j++)

{

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

{

switch(\*(text+j))

{

case 'a':

{

\*(ptr+i) = 1+shift;

break;

}

case 'b':

{

\*(ptr+i) = 2+shift;

break;

}

case 'c':

{

\*(ptr+i) = 3+shift;

break;

}

case 'd':

{

\*(ptr+i) = 4+shift;

break;

}

case 'e':

{

\*(ptr+i) = 5+shift;

break;

}

case 'f':

{

\*(ptr+i) = 6+shift;

break;

}

case 'g':

{

\*(ptr+i) = 7+shift;

break;

}

case 'h':

{

\*(ptr+i) = 8+shift;

break;

}

case 'i':

{

\*(ptr+i) = 9+shift;

break;

}

case 'j':

{

\*(ptr+i) = 10+shift;

break;

}

case 'k':

{

\*(ptr+i) = 11+shift;

break;

}

case 'l':

{

\*(ptr+i) = 12+shift;

break;

}

case 'm':

{

\*(ptr+i) = 13+shift;

break;

}

case 'n':

{

\*(ptr+i) = 14+shift;

break;

}

case 'o':

{

\*(ptr+i) = 15+shift;

break;

}

case 'p':

{

\*(ptr+i) = 16+shift;

break;

}

case 'q':

{

\*(ptr+i) = 17+shift;

break;

}

case 'r':

{

\*(ptr+i) = 18+shift;

break;

}

case 's':

{

\*(ptr+i) = 19+shift;

break;

}

case 't':

{

\*(ptr+i) = 20+shift;

break;

}

case 'u':

{

\*(ptr+i) = 21+shift;

break;

}

case 'v':

{

\*(ptr+i) = 22+shift;

break;

}

case 'w':

{

\*(ptr+i) = 23+shift;

break;

}

case 'x':

{

\*(ptr+i) = 24+shift;

break;

}

case 'y':

{

\*(ptr+i) = 25+shift;

break;

}

case 'z':

{

\*(ptr+i) = 26+shift;

break;

}

}

}

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

{

switch(\*(ptr+i))

{

case 1:

printf("%c",'a');

break;

case 2:

printf("%c",'b');

break;

case 3:

printf("%c",'c');

break;

case 4:

printf("%c",'d');

break;

case 5:

printf("%c",'e');

break;

case 6:

printf("%c",'f');

break;

case 7:

printf("%c",'g');

break;

case 8:

printf("%c",'h');

break;

case 9:

printf("%c",'i');

break;

case 10:

printf("%c",'j');

break;

case 11:

printf("%c",'k');

break;

case 12:

printf("%c",'l');

break;

case 13:

printf("%c",'m');

break;

case 14:

printf("%c",'n');

break;

case 15:

printf("%c",'o');

break;

case 16:

printf("%c",'p');

break;

case 17:

printf("%c",'q');

break;

case 18:

printf("%c",'r');

break;

case 19:

printf("%c",'s');

break;

case 20:

printf("%c",'t');

break;

case 21:

printf("%c",'u');

break;

case 22:

printf("%c",'v');

break;

case 23:

printf("%c",'w');

break;

case 24:

printf("%c",'x');

break;

case 25:

printf("%c",'y');

break;

case 26:

printf("%c",'z');

break;

}

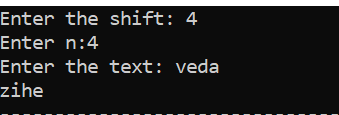
break;

}

}

}

Output:



Explanation:

Once the text is entered , the alphabets are individually converted into their corresponding numbers i.e from 1 to 26 ,later the shift is added to the numbers which are stored in an array , later the new numbers in the array are individually converted into corresponding alphabets.