CAESAR CIPHER

The Caesar cipher is an encryption method used to encrypt plain text to Cipher text. The cipher is done by simply shifting a letter in the alphabet by a certain number of n times (n is called the key) forward for encrypting and shifting backwards for decryption.

The constant n is called a key and is the number of times shifting is done in the algorithm for this cipher.

The C program for the implementation is as explained:

-Variables are being declared ie message (our plain text), ch (temporary variable).
-The plain text is entered and a for loop is used to check every corresponding letter of the alphabet both capital letter and small letter ignoring other characters such as (! @ # $ % ^ . ”) under certain conditions.
The block of code demonstrates all constraints in this program with if statements

BLOCK OF CODE

```
for(i = 0 ; message[i] != '\0' ; i++){
ch= message[i];
//for lowercase letters
if(ch >= 'a' && ch <= 'z'){
ch= ch + key;
if(ch > 'z'){
ch = ch - 'z' + 'a' - 1;
}
message[i]=ch;
}
//for uppercase letters
else if(ch >= 'A' && ch <= 'Z'){
ch = ch + key;
if(ch > 'Z'){
ch = ch - 'Z' + 'A' - 1;
}
message[i] = ch;
}
}
```

ENTIRE C PROGRAM FOR CAESAR CIPHER

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
  char message[100];
  char ch;
  int key;
  int i;

  //start of program
  printf("**********************PLAIN TEXT**********************************\n");
  printf("\n");
  printf("Enter plain text to encrypt: ");
  gets(message);

  printf("\n");
  printf("*****************************KEY************************************\n");
  printf("\n");
  printf("Enter a key: ");
  scanf("%d", &key);

  //the program goes through the plain text
  for(i = 0 ; message[i] != '\0' ; i++){
   ch= message[i];
   //for lowercase letters
   if(ch >= 'a' && ch <= 'z'){
      ch= ch + key;
```

```
        if(ch > 'z'){
            ch = ch - 'z' + 'a' - 1;
        }
        message[i]=ch;
    }
    //for uppercase letters
    else if(ch >= 'A' && ch <= 'Z'){
        ch = ch + key;
        if(ch > 'Z'){
            ch = ch - 'Z' + 'A' - 1;
        }
        message[i] = ch;
    }
   }
   printf("\n");
   printf("\n");
   printf("**************************ENCRYPTED MESSAGE**************************'\n");
   printf("\n");
   printf("Encrypted message: %s",message);
   printf("\n");
   printf("\n");
    printf("**************************END OF
ENCRYPTION**********************************'\n");
return 0;
}
```