Procedure

Algorithm for Caesar Cipher:

Input:

1. A String of lower case letters, called Text.
2. An Integer between 0-25 denoting the required shift.

Procedure:

- Traverse the given text one character at a time .
- For each character, transform the given character as per the rule, depending on whether we're encrypting or decrypting the text.
- Return the new string generated.

1. C Program: Encryption

```c
#include<stdio.h>

 int
main ()
{

char message[100], ch;
  int i, key;

printf ("Enter a message to encrypt: ");
  gets (message);
  printf ("Enter key: ");
  scanf ("%d", &key);

for (i = 0; message[i] != '\0'; ++i)
    {

ch = message[i];
      if (ch >= 'a' && ch <= 'z')
    {
      ch = ch + key;
      if (ch > 'z')
        {

ch = ch - 'z' + 'a' - 1;

}

message[i] = ch;

}

      else if (ch >= 'A' && ch <= 'Z')
    {
      ch = ch + key;
      if (ch > 'Z')
        {

ch = ch - 'Z' + 'A' - 1;

}

message[i] = ch;

}
    }

printf ("Encrypted message: %s", message);
  return 0;
}
```

```
/usr/bin/ld: /tmp/cc0OIub2.o: in function `main':
main.c:(.text+0x37): warning: the `gets' function
Enter a message to encrypt: hello
Enter key: 5
Encrypted message: mjqqt
```

2. C Program: Decryption

```c
#include<stdio.h>

 int
main ()
{

char message[100], ch;
  int i, key;

printf ("Enter a message to decrypt: ");
  gets (message);
  printf ("Enter key: ");
  scanf ("%d", &key);

for (i = 0; message[i] != '\0'; ++i)
    {

ch = message[i];
      if (ch >= 'a' && ch <= 'z')
    {
      ch = ch - key;
      if (ch < 'a')
        {

ch = ch + 'z' - 'a' + 1;

}

message[i] = ch;

}

      else if (ch >= 'A' && ch <= 'Z')
    {
      ch = ch - key;
      if (ch < 'A')
        {

ch = ch + 'Z' - 'A' + 1;

}

message[i] = ch;

}
    }

printf ("Decrypted message: %s", message);
  return 0;
}
```

```
/usr/bin/ld: /tmp/cc22N1EE.o: in function `main':
main.c:(.text+0x37): warning: the `gets' function i
Enter a message to decrypt: mjqqt
Enter key: 5
Decrypted message: hello
```