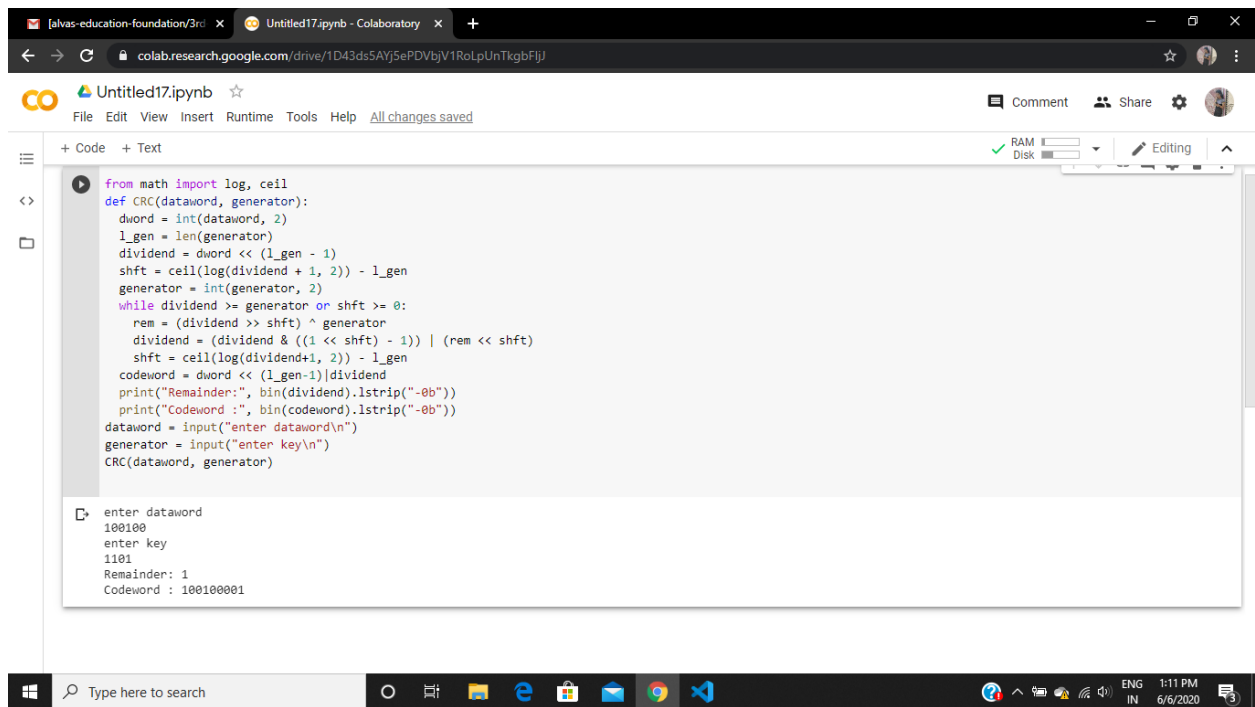


Write a Python program to perform Cyclic Redundancy Check

```
from math import log, ceil
def CRC(dataword, generator):
    dword = int(dataword, 2)
    l_gen = len(generator)
    dividend = dword << (l_gen - 1)
    shft = ceil(log(dividend + 1, 2)) - l_gen
    generator = int(generator, 2)
    while dividend >= generator or shft >= 0:
        rem = (dividend >> shft) ^ generator
        dividend = (dividend & ((1 << shft) - 1)) | (rem << shft)
        shft = ceil(log(dividend+1, 2)) - l_gen
    codeword = dword << (l_gen-1) | dividend
    print("Remainder:", bin(dividend).lstrip("-0b"))
    print("Codeword :", bin(codeword).lstrip("-0b"))
dataword = input("enter dataword\n")
generator = input("enter key\n")
CRC(dataword, generator)
```



The screenshot shows a Google Colab notebook titled 'Untitled17.ipynb'. The code cell contains the same Python program for CRC as shown above. The output cell shows the results of the program execution with the following input and output:

```
enter dataword
100100
enter key
1101
Remainder: 1
Codeword : 10010001
```

The notebook interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help), a toolbar with icons for running, saving, and other actions, and a status bar at the bottom showing system information like RAM, disk space, and the time (1:11 PM, 6/6/2020).