VISHNU UNIVERSAL LEARNING | BVRIT HYDERABAD College of Engineering for Women (UGC-Autonomous)

PPS Lab Activity

Department of CSE Certified that this is a Bonafide Record of the word done by:

Yuvika Sai Simhadri - 23WH1A05G7

Pooja Sangu - 23WH1A05H8

Jeevana Bhavani - 23WH1A05D5

Shreya Pulla - 23WH1A05I4

Of Class CSE C of Year 1 of Semester 1 in PPS Laboratory

Date: ***Signature:***

PROBLEM STATEMENT:

The Atbash cipher, each letter of the plaintext is replaced by its counterpart in the reversed alphabet. Work as secret code The Atbash cipher is a straightforward encryption technique that involves replacing each letter in a message with its corresponding letter from the opposite end of the alphabet. For example, 'A' becomes 'Z,' 'B' becomes 'Y,' and so on. This cipher is symmetrical, meaning the same algorithm can be used for both encoding and decoding. The challenge lies in creating a program or function that effectively applies the Atbash cipher to transform messages securely.

Create a mapping of each letter to its mirror image in the alphabet.

A -> Z, B -> Y, C -> X, and so on.

Iterate through the input message:

For each letter, find its corresponding mirror image based on the mapping.

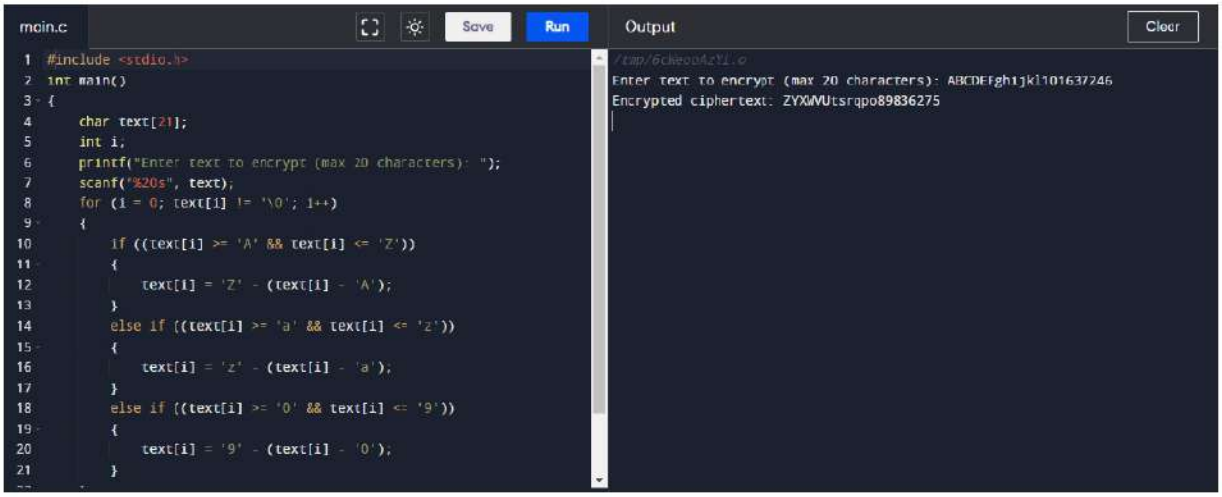
Keep non-alphabetic characters unchanged.

Form the encrypted or decrypted message using the mapped letters.

SOURCE CODE:

```
#include <stdio.h>
int main()
{
    char text[21];
    int i;
    printf("Enter text to encrypt (max 20 characters): ");
    scanf("%20s", text);
    for (i = 0; text[i] != '\0'; i++)
    {
        if ((text[i] >= 'A' && text[i] <= 'Z'))
        {
            text[i] = 'Z' - (text[i] - 'A');
        }
        else if ((text[i] >= 'a' && text[i] <= 'z'))
        {
            text[i] = 'z' - (text[i] - 'a');
        }
        else if ((text[i] >= '0' && text[i] <= '9'))
        {
            text[i] = '9' - (text[i] - '0');
        }
    }
    printf("Encrypted ciphertext: %s\n", text);
    return 0;
}
```

OUTPUT:



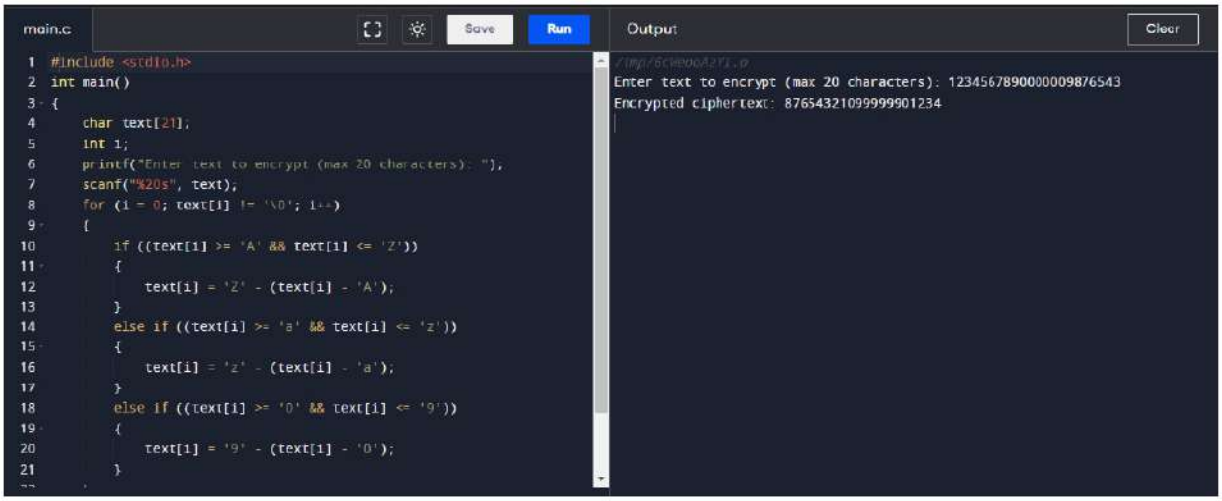
main.c

Save Run

```c
#include <stdio.h>
int main()
{
    char text[21];
    int i;
    printf("Enter text to encrypt (max 20 characters): ");
    scanf("%20s", text);
    for (i = 0; text[i] != '\0'; i++)
    {
        if ((text[i] >= 'A' && text[i] <= 'Z'))
        {
            text[i] = 'Z' - (text[i] - 'A');
        }
        else if ((text[i] >= 'a' && text[i] <= 'z'))
        {
            text[i] = 'z' - (text[i] - 'a');
        }
        else if ((text[i] >= '0' && text[i] <= '9'))
        {
            text[i] = '9' - (text[i] - '0');
        }
```

Output Clear

```
/tmp/6ckeooAzYi.o
Enter text to encrypt (max 20 characters): ABCDEFghijkl101637246
Encrypted ciphertext: ZYXWVUtsrqpo89836275
```



main.c

Save Run

```c
#include <stdio.h>
int main()
{
    char text[21];
    int i;
    printf("Enter text to encrypt (max 20 characters): ");
    scanf("%20s", text);
    for (i = 0; text[i] != '\0'; i++)
    {
        if ((text[i] >= 'A' && text[i] <= 'Z'))
        {
            text[i] = 'Z' - (text[i] - 'A');
        }
        else if ((text[i] >= 'a' && text[i] <= 'z'))
        {
            text[i] = 'z' - (text[i] - 'a');
        }
        else if ((text[i] >= '0' && text[i] <= '9'))
        {
            text[i] = '9' - (text[i] - '0');
        }
```

Output Clear

```
/tmp/6ckeooAzYi.o
Enter text to encrypt (max 20 characters): 1234567890000009876543
Encrypted ciphertext: 8765432109999901234
```