Name: Deepa Kumari

Roll No.: 1814019

Batch: A1

CODE:

ENCRYPTION:

#include<stdio.h>

#include<string.h>

#include<ctype.h>

#include<stdlib.h>

int CalcGCD(int);

int main()

{

int i,j,k,gcd,a,b,numstr[100],numcipher[100];

char str[100],cipher[100];

printf("Enter a string\n");

gets(str);

//converting entered string to Capital letters

for(i=0,j=0;i<strlen(str);i++)

{

if(str[i]!=' ')

{

str[j]=toupper(str[i]);

j++;

}

else

{

str[j]=' ';

j++;

}

}

str[j]='\0';

printf("Entered string is : %s \n",str);

printf("Enter Alpha value in the range of 1 to 25\n");

scanf("%d",&a);

//Checking consitions

if(a<1 || a>25)

{

printf("Invalid alpha input\nPlease Try again!!\n");

exit(0);

}

gcd=CalcGCD(a);

if(gcd!=1)

{

printf("gcd(a,26)=1 but \n gcd(%d,26)=%d\nSorry Try again !\n",a,gcd);

exit(0);

}

printf("Enter Beta value in the range of 0 to 25\n");

scanf("%d",&b);

if(b<0 || b>25)

{

printf("Beta value should lie between 0 and 25\nPlease Try again!!\n");

exit(0);

}

//Conditions Over

//Storing string in terms of ascii and to restore spaces used by -20

for(i=0;i<strlen(str);i++)

{

if(str[i]!=' ')

numstr[i]=str[i]-'A';

else

numstr[i]=-20;

}

//Cipher

/\*If numcipher is more than 25 ,we need to convert it and ensure that it lies in between 0 and 25.

A-0,B-1,C-2,.....Y-24,Z-25\*/

printf("Affine Cipher text is\n");

for(i=0;i<strlen(str);i++)

{

if(numstr[i]!=-20)

{

numcipher[i]=((a\*numstr[i])+b)%26;

printf("%c",(numcipher[i]+'A'));

}

else

{

printf(" ");

}

}

printf("\n");

}

int CalcGCD(int a)

{

int x;

int temp1=a;

int temp2=26;

while(temp2!=0)

{

x=temp2;

temp2=temp1%temp2;

temp1=x;

}

return(temp1);

}

DECRYPTION:

#include<stdio.h>

#include<string.h>

#include<ctype.h>

#include<stdlib.h>

int CalcGCD(int);

int GetMultiplicativeInverse(int);

main()

{

int i,j,k,gcd,alpha,beta,numstr[100],numcipher[100],alphaInverse;

char str[100],cipher[100];

printf("Enter a string\n");

gets(str);

//converting entered string to Capital letters

for(i=0,j=0;i<strlen(str);i++)

{

if(str[i]!=' ')

{

str[j]=toupper(str[i]);

j++;

}

else

{

str[j]=' ';

j++;

}

}

str[j]='\0';

printf("Entered string is : %s \n",str);

printf("Enter Alpha value and must be between 1 and 25 both included\n");

scanf("%d",&alpha);

//Checking conditions

if(alpha<1 || alpha>25)

{

printf("Alpha should lie in between 1 and 25\nSorry Try again !\n");

exit(0);

}

gcd=CalcGCD(alpha);

if(gcd!=1)

{

printf("gcd(alpha,26)=1 but \n gcd(%d,26)=%d\nSorry Try again !\n",alpha,gcd);

exit(0);

}

printf("Enter Beta value and must be between 0 and 25 both included\n");

scanf("%d",&beta);

if(beta<0 || beta>25)

{

printf("Beta value should lie between 0 and 25\nSorry Try again !\n");

exit(0);

}

//Conditions Over

//Program Starts

//Storing string in terms of ascii and to restore spaces I used -20

for(i=0;i<strlen(str);i++)

{

if(str[i]!=' ')

numstr[i]=str[i]-'A';

else

numstr[i]=-20;

}

//For Decryption we need to find multiplicative inverse of Alpha

alphaInverse=GetMultiplicativeInverse(alpha);

printf("MI=%d\n",alphaInverse);

//Deciphering Process

//If numcipher is more than 25 .We need to convert and ensure that lie in between 0 and 25.(indicating Alphabets)

//A-0,B-1,C-2,.....Y-24,Z-25

printf("Affine Cipher text is\n");

for(i=0;i<strlen(str);i++)

{

if(numstr[i]!=-20)

{

numcipher[i]=(alphaInverse\*(numstr[i]-beta))%26;

if(numcipher[i]<0)

{

numcipher[i]=numcipher[i]+26;//To avoid negative numbers

}

printf("%c",(numcipher[i]+'A'));

}

else

{

printf(" ");

}

}

printf("\n");

}

int CalcGCD(int alpha)

{

int x;

int temp1=alpha;

int temp2=26;

while(temp2!=0)

{

x=temp2;

temp2=temp1%temp2;

temp1=x;

}

return(temp1);

}

int GetMultiplicativeInverse(int alpha)

{

int i,MI;

for(i=1;i<=alpha;i++)

{

MI=((i\*26)+1);

if(MI%alpha==0)

{

break;

}

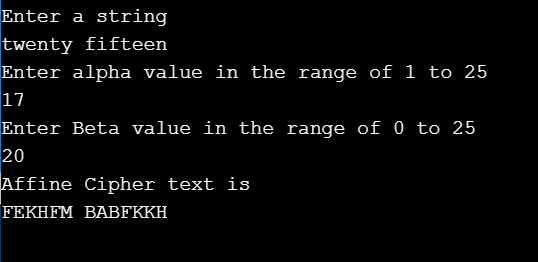
}

MI=MI/alpha;

return(MI);

}

OUTPUT(Encryption):



OUTPUT (Decryption):

