

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

const char LETTRE_POSSIBLE[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 ";

/* ===== FONCTION POUR FAIRE LE CHIFFREMENT ===== */

int trouver_indice(char l)
{
    l = toupper(l);
    for(int i = 0; i<36; i++)
    {
        if(LETTRE_POSSIBLE[i] == l)
        {
            return i;
        }
    }
    return -1;
}

char chiffre_en_caractere(int indice)
{
    if(indice>=0 && indice <36)
    {
        return LETTRE_POSSIBLE[indice];
    }
    return '\\0';
}

int pgcd(int a, int b)
{
    while(b!= 0)
    {
        int temp = a %b;
        a = b;
        b = temp;
    }
    return a;
}

/* == FONCTION POUR CALCULER LE DETERMINANT == */

void sous_matrice(int **matrice , int **sous_matrice, int ligne, int
colonne, int ordre)
{
    int r = 0;
    int c = 0;
    for(int i = 0; i < ordre; i++)
    {
        if(i == ligne)
        {
            continue;
        }else{
            c = 0;
        }
    }

```

```

        for(int j = 0; j < ordre ; j++)
        {
            if(j == colonne)
            {
                continue;
            }
            sous_matrice[r][c++] = matrice[i][j];
        }
        r++;
    }
}

int calcul_determinanr(int **matrice , int ordre)
{
    if(ordre == 1)
    {
        return matrice[0][0];
    }
    else if(ordre == 2)
    {
        return matrice[0][0] * matrice[1][1] - matrice[0][1] *
matrice[1][0];
    }

    int det = 0;
    int sous_matriX = malloc((ordre-1) * sizeof(int));
    for(int i = 0; i < ordre; i++)
    {
        sous_matriX[i] = malloc((ordre-1) * sizeof(int))
    }

    for(int col = 0 ; col < ordre; col++)
    {
        sous_matrice(matrice, sous_matriX, 0, col, ordre);
        int signe = (col % 2 == 0) ? 1 : -1;
        det += signe * matrice[0][col] * calcul_determinanr(sous_matriX,
ordre - 1);
    }
    for(int i = 0; i < ordre-1; i++)
    {
        free(sous_matriX[i]);
    }
    free(sous_matriX);
    return det;
}

void chiffrement_HILL(char *message, int **K , int ordre)
{
    int longueur = strlen(message);
    while(longueur % ordre != 0)
    {
        message[longueur++] = ' ';
        message[longueur] = '\0';
    }
    printf("\nTexte chiffré : ");

```

```

for(int i = 0; i < ordre; i++)
{
    int bloc[ordre], resultat[ordre];
    for(int j = 0; j < ordre; j++)
    {
        bloc[j] = trouver_indice(message[i + j]);
    }
    for(int r = 0; r < ordre; r++)
    {
        resultat[r] = 0;
        for(int c = 0; c < ordre; c++)
        {
            resultat[r] += K[r][c] * bloc[c];
        }
        resultat[r] %= 36;
    }
    for(int j = 0; j < ordre; j++)
    {
        printf("%c", chiffre_en_caractere(resultat[j]));
    }
    printf("\n")
}

}

int main()
{
    int ordre;
    printf("Entrez l'ordre de la matrice : ");
    scanf("%d", &ordre);

    if(ordre < 2 || ordre > 8)
    {
        printf("L'ordre de la matrice doit être compris entre 2 et
8.\n");
        return 1;
    }
    int **K = malloc(ordre * sizeof(int *));
    for(int i = 0; i < ordre; i++)
    {
        K[i] = malloc(ordre * sizeof(int));
    }
    printf("Entrez la matrice de clé (%d x %d) :\n", ordre, ordre);
    for(int i = 0; i < ordre; i++)
    {
        for(int j = 0; j < ordre; j++)
        {
            scanf("%d", &K[i][j]);
        }
    }
    int det = calcul_determinant(K, ordre);
    printf("Le déterminant de la matrice est : %d\n", det);

    int pg = pgcd(abs(det), 36);
    printf("Le pgcd de %d et 36 est : %d\n", abs(det), pg);

    if(det == 0 || pg != 1)
    {
        printf("La matrice n'est pas inversible modulo 36.\n");
    }
}

```

```
        return 1;
    }
    while(getchar() != '\n');
    char message[100];
    printf("Entrez le message à chiffrer : ");
    fgets(message, sizeof(message), stdin);
    message[strcspn(message, "\n")] = '\0';
    printf("Message original : %s\n", message);
    chiffrement_HILL(message, K, ordre);
    printf("Texte chiffré : %s\n", message);
    for(int i = 0; i < ordre; i++)
    {
        free(K[i]);
    }
    free(K);

    return 0;
}
```