

```

#include <stdio.h>
#include <math.h>
// #include <stdlib.h>
// #include <string.h>
// #define CEND '\0' le caractère de fin rajouté par défaut
#define N 10

```

```

void _strcpy(char s1[],char s2[]){
    int i=0;
// while (s2[i++]==s1[i])
    while (s1[i]!='\0'){
        s2[i]=s1[i];
        i++;
    }
    s2[i]='\0';
}

```

```

int min (int a,int b){
    return (a>b ? b : a);
}

```

```

int strlen (char s[]){
    int i=0;
    for (i=0; s[i]!='\0';i++){
// printf(" val %c\n",s[i]);
// while (s[i]) {i++;}
    }
    printf(" val %c",s[i])=="rien"
    return i;
}

```

```

void strconcat(char s1[],char s2[], char s3[]){
    int i=0;
    for (i=0;s1[i];i++){
        s3[i]=s1[i];
    }
    int j=0;
    for (j=0;s2[j];j++){
        s3[j]=s2[j];
    }
    s3[i+j]='\0';
}

```

```

//nb d'apparitions de S2 dans S1
int __strsub(char s1[],char s2[]){
    int i,j,nb;
    nb=0;
    for (i=0;s1[i];i++){
        j=0;
        while (s1[i+j] && s2[j] && s1[i+j]==s2[j]){
            j++;
        }
        if (s2[j]=='\0'){
            nb++;
        }
    }
    return nb++;
}

```

```

// insertion de s2 dans s1 a la position p
void strinsert(char s1[],char s2[],int p){
    int i,q,j,l1,l2;

    l1=strlength(s1);
    q=min(p,l1);
    printf ("la valeur q=min() %d\n",q);
    l2=strlength(s2);
// decalage
    for (i=l1;i>=q;i--){
        printf ("decalage i=%d\n",i);
        s1[i+l2]=s1[i];
    }
//copie

```

```

for (i=0;i<12;i++){
    s1[q+i]=s2[i];
    printf ("copie i=%d\n",i);
}
return;
//
}

```

```

int main(int argc, char *argv[])
{

```

```

    char s[20];
    printf("rentrez une chaine de caracteres \n");
    scanf("%s",s);
    printf ("la longueur de la chaine -> %s <- est %d
\n",s,strlen(s));
// printf("min %d",min(12,6));
    char s1[20];
    printf("rentrez une chaine de caracteres \n");
    scanf("%s",s1);
    printf ("la longueur de la chaine -> %s <- est %d
\n",s1,strlen(s1));

```

```

    strinsert(s,s1,3);
    printf ("la longueur de la nouvelle chaine -> %s <- est %d
\n",s,strlen(s));

```

```

    printf("nb fois ds la chaine %d",__strsub(s,s1));

```

```

system("PAUSE");
return 0;
}

```

```

Nb voyelles
#include <stdio.h>
#include <math.h>
// #include <stdlib.h>
// #include <string.h>
#define CEN '\0' le caractère de fin rajouté par défaut
#define N 10
#define TRUE 1
#define FALSE 0

int testVoyelle(char* s){
    if (*s=='a' || *s=='e' || *s=='i' || *s=='o' || *s=='u' ||
*s=='y'){
        return TRUE;
    }
    else {
        return FALSE;
    }
}

int nbVoyelles(char* c){
    int nb=0;
    char *s;

    for (s=c; *s!='\0' ;s++){
        if (testVoyelle(s)){
            nb++;
        }
    }
    return nb;
}

int main(int argc, char *argv[])
{

    char s[20];
    printf("rentrer une chaine de caracteres \n");
    scanf("%s",s);
    printf ("le nombre de voyelles dans la chaine -> %s <- est
egale a %d \n",s,nbVoyelles(s));

    system("PAUSE");
    return 0;
}

```

```

Prefix
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
// #include <string.h>
#define CEN '\0' le caractère de fin rajouté par défaut
#define N 10
#define TRUE 1
#define FALSE 0

int min (int a,int b){
    return (a>b ? b : a);
}

int strlenht (char s[]){
    int i=0;
    for (i=0; s[i]!='\0';i++){
        // printf(" val %c\n",s[i]);
    }
    // while (s[i]){i++;}
    // printf(" val %c",s[i])=="rien"
    return i;
}

int prefix(int p,char* str,char * strp){

    int p1 = min(p,strlenht(str));
    int i=0;
    if (p>p1) {
        printf("On ne copie que %d\n caracteres",p1);
    }
    printf("dans prefix : on recopie la lettre : \n");
    while (i<p1){
        printf("t%c",str[i]);
        strp[i]=str[i];
        i++;
    }
    printf("\n");
    strp[i]= '\0';
    return i;
}

int prefixcom(char s1[],char s2[],char s3[]){

    int i=0;
    printf("dans prefixcom : on recopie la lettre : \n");
    while (s1[i] && s2[i] && s1[i]==s2[i]){

        s3[i]=s1[i];
        printf("%c\t",s1[i]);
        i++;
    }
    s3[i]='\0';
    return i;
}

void lire(char *s){
    int i=0;
    for (i=0; s[i]!='\0';i++){
        printf("%c",s[i]);
    }
    printf("\n");
}

int main(int argc, char *argv[])
{
    int nb;
    char strp[20];

    char s[20];
    char s3[20];
    printf("rentrer une chaine de carateres \n");
    scanf("%s",s);
    // lire(s);

    printf("combien d'elements a recopier\n");
    scanf("%d",&nb);
    prefix(nb,s,strp);
    printf(" \n voici la liste recopiee \t");
    lire(strp);
    prefixcom(s,strp,s3);
    printf(" \n voici le prefix commun \t");
    lire(s3);

    system("PAUSE");
    return 0;
}

```

```

Code caché
#include <stdio.h>
#include <stdlib.h>

#define DIMMAX 10 // pas utilise ici
#define TRUE 1
#define FALSE 0

int strlenht (char s[]){
    int i=0;
    for (i=0; s[i]!='\0';i++){
    }
    return i;
}

int hide(char *str1, char *str2){

    int pos_recherche=0;
    int indice_code=0;
    int trouve= TRUE;
    int dim_code;
    int dim_tab;
    int val_cherchee;

    dim_code=strlenht(str1);
    dim_tab=strlenht(str2);
// on peut ici tester que str1<str2
    if (dim_code>dim_tab){
        printf("Merci de verifier votre recherche");
        return FALSE;
    }

    while (trouve && indice_code<dim_code){

        trouve= FALSE; // je n'ai pas encore trouve l'ele
        val_cherchee=str1[indice_code];
        // printf("la valeur recherche = %d\n",val_cherchee);
        while (!trouve && pos_recherche<dim_tab){
            if (str2[pos_recherche]==val_cherchee){
                printf("== trouve en position %d\n\n",pos_recherche);
                trouve = TRUE; //permet de passer à l'element suivant
            }
            pos_recherche++; // pour faire avancer la boucle
            if (!trouve){
                printf("----- en position
%d\n",pos_recherche);
            }
        }
        indice_code++; // recherche l'element suivant
// affichage resultat intermediaire
        if (!trouve){
            printf(" la valeau\n\n");
        }
    }
    return(trouve);
}

int main(int argc, char *argv[])
{

//-----définition d'un tableau test-----
char phrase[20];
printf("rentrer une chaine de carateres \n");
scanf("%s",phrase);
char code[20];
printf("rentrer le code a trouver dans la chaine de carateres
\n");
scanf("%s",code);

//-----Affiche si trouve ou pas
if ( hide(code,phrase)){
        printf ("\n\nBILAN / le code est present\n\n\n");
    }
else{
        printf ("\n\nBILAN / le code n'est pas present\n\n\n");
    }

//-----
system("PAUSE");
return 0;
}

```

```

Le javanais
#include <stdio.h>
#include <math.h>
// #include <stdlib.h>
// #include <string.h>
#define CEN '\0' le caractère de fin rajouté par défaut
#define N 10
#define TRUE 1
#define FALSE 0

int min (int a,int b){
    return (a>b ? b : a);
}

int strlenht (char s[]){
    int i=0;
    for (i=0; s[i]!='\0';i++){
    }
    return i;
}

int testVoyelle(char* s){
    if (*s=='a' || *s=='e' || *s=='i' || *s=='o' || *s=='u' ||
*s=='y'){
        return TRUE;
    }
    else {
        return FALSE;
    }
}

void fairecode(char *s,char *code){
    code[0]='d';
    code[1]=*s;
    code[2]='g';
    code[3]=*s;
    code[4]='\0';
return;
}

// insertion de s2 dans s1 a la position p
void strinsert(char s1[],char s2[],int p){
    int i,q,j,l1,l2;

    l1=strlength(s1);
    q=min(p,l1);
    // printf ("la valeur q=min() %d\n",q);
    l2=strlength(s2);
    // decalage
    for (i=l1;i>=q;i--){
    // printf ("decalage i=%d\n",i);
        s1[i+l2]=s1[i];
    }
    //copie
    for (i=0;i<l2;i++){
        s1[q+i]=s2[i];
    // printf ("copie i=%d\n",i);
    }
    return;
}

void lire(char *s){
    int i=0;
    for (i=0; s[i]!='\0';i++){
        printf("%c",s[i]);
    }
    printf("\n");
}

void java(char* c){

```

```

int position;
char *s;
char *tmp;
char code[4];

for (s=c,position=0; *s!='\0' ;s++,position++){
    if (testVoyelle(s)){
        fairecode(s,code);
        // lire(code);
        strinsert(c,code,position+1);
        // lire(c);
        // lire(s);
        s=s+4;
        position=position+4;
    }
}

int main(int argc, char *argv[])
{

    char s[20];
    printf("rentrer une chaine de caracteres \n");
    scanf("%s",s);
    java(s);
    printf("le code est %s\n",s);

    system("PAUSE");
    return 0;
}

```