

Exercice 1 :

1.

```
#include <stdio.h>
main()
{
    char M1[30], M2[30], M3[30], M4[30], M5[30];
    printf("Entrez 5 mots, séparés par des espaces :\n");
    scanf ("%s %s %s %s %s", M1, M2, M3, M4, M5);
    printf("%s %s %s %s %s\n",M5, M4, M3, M2, M1);
    return 0;
}
```

2.

```
#include <stdio.h>
main()
{
    /* Déclarations */
    char TXT[201]; /* chaîne donnée      */
    int I, J;        /* indices courants */

    /* Saisie des données */
    printf("Entrez une ligne de texte (max.200 caractères) :\n");
    gets(TXT);

    /* Eliminer les lettres 'e' et comprimer : */
    /* Copier les caractères de I vers J et incrémenter J */
    /* seulement pour les caractères différents de 'e'. */
    for (J=0, I=0 ; TXT[I] ; I++)
    {
        TXT[J] = TXT[I];
        if (TXT[I] != 'e') J++;
    }

    /* Terminer la chaîne !! */
    TXT[J] = '\0';
    /* Edition du résultat */
    puts(TXT);
    return 0;
}
```

3.

```
#include <stdio.h>
#include <string.h>

main() {
/* Déclarations */

char MOT[10][50]; /* tableau de 10 mots à trier */
char AIDE[50]; /* chaîne d'aide pour la permutation */
int I; /* rang à partir duquel MOT n'est pas trié */
int J; /* indice courant */

int PMOT; /* indique la position du prochain mot dans la suite lexicographique.
*/
/* Saisie des données */

for (J=0; J<10; J++) {
printf("Mot %d : ", J);
gets(MOT[J]); /* ou : scanf ("%s\n", MOT[J]); */
}

/* Tri du tableau par sélection directe du prochain mot dans la suite
lexicographique. */

for (I=0; I<9; I++) {
/* Recherche du prochain mot*/
PMOT=I;
for (J=I+1; J<10; J++)
if (strcmp(MOT[J], MOT[PMOT]) < 0)
PMOT=J;
/* Echange des mots à l'aide de strcpy */
strcpy(AIDE, MOT[I]);
strcpy(MOT[I], MOT[PMOT]);
strcpy(MOT[PMOT], AIDE); }
/* Edition du résultat */
```

```

printf("Tableau trié lexicographiquement :\n");
for (J=0; J<10; J++)
puts(MOT[J]); /* ou : printf("%s\n",MOT[J]); */
printf("\n");
return (0); }
```

4.

```

#include <stdio.h>
main()
{
    int N;
    char JOUR[8][9] = { "Erreur!", "lundi", "mardi", "mercredi",
                        "jeudi", "vendredi", "samedi", "dimanche" };

    printf("Entrez un nombre entre 1 et 7 : ");
    scanf("%d", &N);
    /* Affichage du résultat */
    if (N>0 && N<8)
        printf("Le %de%c jour de la semaine est %s.\n",
               N, (N==1)?' ':' ', JOUR[N]);
    else
        puts(JOUR[0]);
    return 0;
}
```

Exercice 2:

Trouvez les erreurs dans les suites d'instruction suivantes :

1) **short *p , x = 34; *p = x;**

***p = x** est incorrect parce que le pointeur p n'est pas initialisé

2) **long x = 17 , *p = x; *p = 17;**

***p = x** est incorrect. Pour que p pointe sur x à la déclaration, on écrit : ***p = &x**

3) **double *q; long x = 17 , *p = &x; q = p;**

q = p incorrect. q et p deux pointeurs sur des types différent

4) **short x, *p; &x = p;**

&x = p incorrect. &x n'est pas une variable (**lvalue**) et par conséquent l'expression **&x** ne peut pas figurer à gauche d'une affectation.

5) **char mot[10], car = 'A', *pc = &car ; mot = pc;**

mot = pc incorrect.

mot est un pointeur constant (nom d'une variable tableau) par conséquent on ne peut pas changer sa valeur.

Exercice 3 :

```
#include <stdio.h>
main()
{
    int A[50];
    int N;
    int X;
    int *P1, *P2;

    printf("Dimension du tableau (max.50) : ");
    scanf("%d", &N );
    for (P1=A; P1<A+N; P1++)
    {
        printf("Elément %d : ", P1-A);
        scanf("%d", P1);
    }
    printf("Introduire l'élément X à éliminer du tableau : ");
    scanf("%d", &X );
    /* Affichage du tableau */
    for (P1=A; P1<A+N; P1++)
        printf("%d ", *P1);
    printf("\n");

    /* Effacer toutes les occurrences de X et comprimer : */
    /* Copier les éléments de P1 vers P2 et augmenter P2 */
    /* pour tous les éléments différents de X. */

    for (P1=P2=A; P1<A+N; P1++)
    {
        *P2 = *P1;
        if (*P2 != X)
            P2++;
    }
    /* Nouvelle dimension de A */
    N = P2-A;

    /* Résultat */
    for (P1=A; P1<A+N; P1++)
        printf("%d ", *P1);
    printf("\n");
    return 0;
```

```
}
```

Exercice 4 :

a) en utilisant uniquement le formalisme tableau

```
#include <stdio.h>

main() {
    /* Déclarations */
    char CH[101];
    /* chaîne donnée */
    int I,J;
    /* indices courants */
    int PALI;
    /* indicateur logique : */
    /* vrai si CH est un palindrome */
    /* Saisie des données */
    printf("Entrez une ligne de texte (max.100 caractères) :\n");
    gets(CH);
    /* Placer J sur la dernière lettre de la chaîne */
    for(J=0; CH[J]; J++) /* J=0; While( CH[J] ) J++; J--; */
    J--;
    /* Contrôler si CH est un palindrome */
    PALI=1;
    for (I=0 ; PALI && I<J ; I++,J--)
        if (CH[I] != CH[J])
            PALI=0;
    /* Affichage du résultat */
    if (PALI)
        printf("La chaîne \"%s\" est un palindrome.\n", CH);
    else
        printf("La chaîne \"%s\" n'est pas un palindrome.\n", CH);
    return (0); }
```

b) en utilisant des pointeurs au lieu des indices numériques :

```
#include <stdio.h>
```

```

main() {
    /* Déclarations */
    char CH[101];
    /* chaîne donnée */
    char *P1, *P2;
    /* pointeurs d'aide */
    int PALI;
    /* indicateur logique : */
    /* vrai si CH est un palindrome */
    /* Saisie des données */
    printf("Entrez une ligne de texte (max.100 caractères) :\n");
    gets(CH);
    /* Placer P2 sur la dernière lettre de la chaîne */
    for (P2=CH; *P2; P2++) ;
    P2--;
    /* Contrôler si CH est un palindrome */
    PALI=1;
    for (P1=CH ; PALI && P1<P2 ; P1++,P2--)
        if (*P1 != *P2)
            PALI=0;
    /* Affichage du résultat */
    if (PALI)
        printf("La chaîne \"%s\" est un palindrome.\n", CH);
    else
        printf("La chaîne \"%s\" n'est pas un palindrome.\n", CH);
    return (0);
}

```

Exercice 5:

```

#include <stdio.h>

main() {
    /* Déclarations */

    int A[50][50]; /* matrice */
    int N, M, *pt; /* dimensions de la matrice */
    int I, J; /* indices courants */
    pt=&A[0][0];
    /* Saisie des données */
    printf("Nombre de lignes (max.50) : ");
    scanf("%d", &N );
    printf("Nombre de colonnes (max.50) : ");
    scanf("%d", &M );
    /* Lecture de la matrice au clavier */
    for (I=0; I<N; I++) {
        for (J=0; J<M; J++) {
            printf("Elément[%d][%d] : ",I,J);
            scanf("%d", pt+i*50+j); }
        }
    /* a) Affichage de la matrice */
    printf("Matrice donnée :\n");
    for (I=0; I<N; I++) {
        for (J=0; J<M; J++)
            printf("%7d", *(pt+i*50+j));
        printf("\n"); }

    /* b) Affichage de la transposée de A */
    printf("Matrice transposée :\n");
    for (J=0; J<M; J++) {
        for (I=0; I<N; I++)
            printf("%7d ", *(pt+i*50+j));
        printf("\n"); }

    /* c) Interprétation de la matrice comme vecteur : */
    for (I=0; I<N; I++)

```

```
for (J=0; J<M; J++)
printf(" %d ", *(pt+i*50+j));
printf("\n");
return (0); }
```