

Exercice 1 * Construire une structure ELEVE qui contiendra comme champs le prénom de l'élève, sa date de naissance ainsi que la note qu'il a obtenue a son dernier partiel d'info (entre A et E). Deux élèves s'inscrivent en fournissant tous les renseignements nécessaires. L'ordinateur doit ensuite donner le prénom de celui qui est le plus âgé, et de celui qui a obtenu la meilleure note.

Solution 1 #include <stdio.h>

```
typedef char PRENOM[10]; // le prenom est une chaine de caractere (max 10)

typedef int DATE[3]; // la date est constituee de 3 entiers

typedef char NOTE; // la note est une lettre entre A et E

typedef struct
{
    PRENOM prenom;
    DATE date;
    NOTE note;
} ELEVE;

main()
{
    /**Declarations */
    ELEVE e1, e2; /* les deux eleves */

    /*** Premier eleve */
    printf("Saisie du premier eleve\n");
    /* Saisie du prenom */
    printf("Quel est votre prenom?:\n");
    scanf("%s", e1.prenom);
    /* Saisie de la date de naissance */
    printf("Quelle est votre date de naissance?:\n");
    do
    {
        printf("Jour:");
        scanf( "%d", &(e1.date[0]));
    }
    while((e1.date[0]>31)|| (e1.date[0]<1)); //teste la validite du jour
    do
    {
        printf("Mois:");
        scanf( "%d", &(e1.date[1]));
    }
    while((e1.date[1]>12)|| (e1.date[0]<1)); //teste la validite du mois

    do
    {
        printf("Annee:");
        scanf( "%d", &(e1.date[2]));
    }
    while((e1.date[2]>2000)|| (e1.date[2]<1900)); //teste la validite de l'annee
    /* Saisie de la note */
    do
    {
        printf("Quelle note avez vous obtenue au partiel:");
        fflush(stdin); //ca Marche mieux avec un fflush...
        scanf( "%c", &(e1.note));
    }
    while( (e1.note>'E') || (e1.note<'A')); //teste la validite de la note
    /*** Deuxieme eleve */
    printf("Saisie du deuxieme eleve\n");
```

```

/* Saisie du prenom */
printf("Quel est votre prenom?:\n");
scanf("%s",e12.prenom);
/* Saisie de la date de naissance */
printf("Quelle est votre date de naissance?:\n");
do
{
printf("Jour:");
scanf( "%d",&(e12.date[0]));
}
while((e12.date[0]>31)|| (e12.date[0]<1)); //teste la validite du jour
do
{
printf("Mois:");
scanf( "%d",&(e12.date[1]));
}
while((e12.date[1]>12)|| (e12.date[0]<1)); //teste la validite du mois
do
{
printf("Annee:");
scanf( "%d",&(e12.date[2]));
}
while((e12.date[2]>2000)|| (e12.date[2]<1900)); //teste la validite de l'annee
/* Saisie de la note */
do
{
printf("Quelle note avez vous obtenue au partiel:");
fflush(stdin); //ca Marche mieux avec un fflush...
scanf( "%c",&(e12.note));
}
while( (e12.note>'E') || (e12.note<'A')); //teste la validite de la note
*** Comparaison des deux eleves */
/* comparaison de la date */
for (int i=2;i>=0;i=i-1)
{
if (e11.date[i]!=e12.date[i])
{
if (e11.date[i]>e12.date[i])
{
printf("%s est le plus age\n",e12.prenom);
}
else
{
printf("%s est le plus age\n",e11.prenom);
}
break;
}
}
/* Comparaison des notes */
if (e11.note<e12.note)
{
printf("%s a eu la meilleure note\n",e11.prenom);
}
else
{
printf("%s a eu la meilleure note\n",e12.prenom);
}
return 0;
}

```

Exercice 2 *** Pour Noel, vous souhaitez acheter un téléviseur dernier cri pour votre professeur préféré. Chez CARROUF, toujours en avance d'une mode, c'est un ordinateur qui va vous aider à choisir votre cadeau en fonction de la somme que vous possédez.*

Créez une structure décrivant un téléviseur, ayant pour champs : la taille (35,45,70 cm), l'affichage (couleur ou N&B), l'écran (plat ou non), la marque (SYNO, SIMSON, ou Tournotour). Chacune de ces caractéristiques aura une contribution dans le prix final du téléviseur qui seront définies comme constantes (PRIX_COUL=100 et PRIX_NB=50 par

exemple).

Vous réaliserez ensuite un programme qui vous demande de quelle somme vous disposez, ainsi que les fonctionnalités que vous désirez, et vous dira si la machine de vos rêves est dans vos moyens.

Solution 2 L'exercice n'est pas difficile, sa difficulté reside dans la longueur des saisies et l'utilisation rusée des structures. L'utilisation des constantes dans le prix rallonge considérablement la solution. Une solution élégante consiste à créer des énumérations pour chaque caractéristique, ainsi que des tableaux d'entiers contenant les prix correspondant à chaque caractéristique.

```
#include <stdio.h>
typedef enum {S,M,B} TAILLE;
typedef enum {NB,RGB} COULEUR;
typedef enum {PLAT,STANDARD} ECRAN;
typedef enum {SYNO,SIMSON,TOURNOTOUR} MARQUE;
typedef struct
{
    TAILLE taille;
    COULEUR couleur;
    ECRAN ecran;
    MARQUE marque;
} TV;

main()
{
    /*** Table des prix*/
    int prix_taille[]={50,100,200};
    int prix_couleur[]={50,100};
    int prix_ecran[]={100,50};
    int prix_marque[]={50,60,70};

    TV tv;//televiseur
    int somme_init,prix=0;//somme initiale et prix du televiseur
    int OK=0;//Booleen achat possible

    /*** Caracteres de saisie clavier */
    int taille;
    char coul;
    char ecran;
    char marque;

    /* Saisie de la somme initiale*/
    printf("Bonjour, de quelle somme disposez-vous?\n");
    scanf("%d",&somme_init);

    while(!OK)//tant qu'on ne dispose pas d'assez d'argent
    {
        /*Saisie de la taille*/
        do{
            printf("Quelle taille desirez vous?(35,45,70)\n");
            scanf("%d",&taille);
        }
        while ((taille!=35)&&(taille!=45)&&(taille!=70));
        /* Enregistrement de la taille */
        switch (taille)
        {
            case (35):tv.taille=S;break;
            case (45):tv.taille=M;break;
            case (70):tv.taille=B;break;
        }

        /*Saisie de la couleur*/
        do{
            printf("Quelle couleur desirez vous?((N)B,(R)VB)\n");
            fflush(stdin);
            scanf("%c",&coul);
        }
        while ((coul!='N')&&(coul!='R'));
```

```

/* Enregistrement de la couleur */
switch (couleur)
{
    case 'N':tv.couleur=NB;break;
    case 'R':tv.couleur=RGB;break;
}

/*Saisie du type d'ecran*/
do{
    printf("Quelle ecran desirez vous?(P)LAT,(S)TANDARD\n");
    fflush(stdin);
    scanf("%c",&ecran);
}
while ((ecran!='P')&&(ecran!='S'));
/* Enregistrement du type d'ecran */
switch (ecran)
{
    case 'P':tv.ecran=PLAT;break;
    case 'S':tv.ecran=STANDARD;break;
}

/*Saisie de la marque*/
do{
    printf("Quelle marque desirez vous?(S(Y)NO,(S)IMSON,(T)OURNOTOUR\n");
    fflush(stdin);
    scanf("%c",&marque);
}
while ((marque !='Y')&&(marque!='S')&&(marque!='T'));
/* Enregistrement de la marque */
switch (marque)
{
    case 'Y':tv.marque=SYNO;break;
    case 'S':tv.marque=SIMSON;break;
    case 'T':tv.marque=TOURNTOUR;break;
}

/** Calcul du prix */
prix=prix_taille[tv.taille]+prix_couleur[tv.couleur]+prix_ecran[tv.ecran]+prix_marque[tv.marque];
printf("le prix de ce televiseur est %d F, ",prix);
if (prix<somme_init)
{
    printf("vous pouvez l'acheter!\n");
    OK=1;
}
else
{
    printf("desole, c'est top cher pour vous...\n");
}
} //fin while(!OK)

printf("Au revoir!\n");
return 0;
}

```

Exercice 3 Conjugaison **

En utilisant la concaténation vue à l'exercice 11.2, réaliser un programme qui conjugue au présent un verbe du premier groupe donné par l'utilisateur.

On utilisera deux tableaux de chaînes de caractères, l'un contenant la liste des pronoms personnels, l'autre contenant la liste des terminaisons correspondantes.

Solution 3

```

#include <stdio.h>
#include <string.h>
#define TAILLE_VERB_MAX 20
#define TAILLE_AFFI_MAX 30

int main()

```

```

{
/**** Dclarations */

char PRONOM[6][6]={"je ", "tu ", "il ", "nous ", "vous ", "ils "};/* Tableau des pronoms */
char TERM[6][4]={"e","es","e","ons","ez","ent"};/* Terminaison a rajouter a la fin du verbe */
char VERB[TAILLE_VERB_MAX]; /* chane contenant le verbe */
char AFFI[TAILLE_AFFI_MAX]; /* chane pour l'affichage */
int L; /* longueur de la chane VERB*/
int i,j,k;/* compteurs*/

/**** Saisie des donnes */
printf("Verbe : ");
scanf("%s",VERB);

/**** Contrler s'il s'agit d'un verbe en 'er' */

/* On calcule la taille de VERB */
L=0;
char c=VERB[L];
while (c!='\0')
    {L++;c=VERB[L];}

if ((VERB[L-2]!='e') || (VERB[L-1]!='r'))
    printf("%s","\aCe n'est pas un verbe du premier groupe!");
else
    {
/**** On parcourt tous les pronoms */

for (i=0;i<6;i++)
    {
/* Couper la terminaison 'er'. */
VERB[L-2]='\0';

/* Conjuguer ... */

/*Pronom*/
AFFI[0]='\0';
j=0;k=0;c=PRONOM[i][j];
while(c!='\0')
    {
AFFI[k]=c;
j++;k++;
c=PRONOM[i][j];
    }
/* Racine*/
j=0;c=VERB[j];
while(c!='\0')
    {
AFFI[k]=c;
j++;k++;
c=VERB[j];
    }
/*Terminaison*/
j=0;c=TERM[i][j];
while(c!='\0')
    {
AFFI[k]=c;
j++;k++;
c=TERM[i][j];
    }
/* Fin de chaine*/

AFFI[k]='\0';
printf("%s\n",AFFI);
    }
}
}

```

```
return 0;  
}
```

Exercice 4 Tri à bulle **

On a vu dans l'exercice 11.4 une procédure de tri par sélection du minimum. Nous allons voir une procédure plus évoluée.

Tri par propagation (bubble sort) : En recommençant chaque fois au début du tableau, on effectue à plusieurs reprises le traitement suivant: on propage, par permutations successives, le plus grand élément du tableau vers la fin du tableau (comme une bulle qui remonte la surface d'un liquide). On arrête le traitement lorsqu'il n'y a plus de permutation possible.

On créera un indice FIN indiquant l'endroit où a eu lieu la dernière permutation, afin de ne plus traiter les parties du tableau déjà triées.

Tant que le tableau n'est pas trié

 Pour l allant de 0 a FIN

 Si $TAB[l] > TAB[l+1]$,

 on les permute.

 FIN=l

 Finsi

finpour

fintantque

Exemple :

```
3 2 1 4
premier balayage
2 3 1 4, FIN=0
2 1 3 4, FIN=1
second balayage
1 2 3 4, FIN=0
fin
```

Écrire en C un algorithme de tri par propagation sur un tableau de caractères.

Solution 4

```
#include <stdio.h>
#define TAILLE_MAX 50
int main()
{
    /* Dclarations */
    char A[TAILLE_MAX]; /* tableau donn */
    int N; /* dimension */
    int I; /* rang partir duquel A est tri */
    int J; /* indice courant */
    char SWAP; /* pour la permutation */
    int FIN; /* position o la dernire permutation a eu lieu. */
    /* permet de ne pas trier un sous-ensemble dj tri. */
    /* Saisie des donnees */
    printf("Dimension du tableau (max.%d) : ",TAILLE_MAX);
    scanf("%d", &N );
    for (J=0; J<N; J++)
    {
        printf("Element %d : ", J);
        scanf("%s", &A[J]);/*!!!!%c et get char ne marchent pas car
        ils prennent en compte le retour a la ligne */
    }
    /* Affichage du tableau */
    printf("Tableau donn :\n");
    for (J=0; J<N; J++)
        printf("%c ", A[J]);
    printf("\n");
    /* Tri du tableau par propagation de l'lement maximal. */
    for (I=N-1 ; I>0 ; I=FIN)
    {
        FIN=0;
        for (J=0; J<I; J++)
            if (A[J]>A[J+1])
            {
                FIN=J;
                SWAP=A[J];
                A[J]=A[J+1];
                A[J+1]=SWAP;
            }
    }
    /* Edition du resultat */
    printf("Tableau tri :\n");
    for (J=0; J<N; J++)
        printf("%c ", A[J]);
    printf("\n");
    return 0;
}
```

Exercice 5 Application : Tri d'une liste de mots ***

En utilisant l'ordre lexicographique défini à l'exercice 11.3, et l'algorithme de tri défini à l'exercice précédent, réaliser un programme qui classe, selon l'ordre du dictionnaire, une liste de mots fournis par l'utilisateur. Comparer la rapidité des différents algorithmes de tri.

Solution 5

```
#include <stdio.h>
#define TAILLE_MAX 50
#define TAILLE_MAX_CHAINE 10
int main()
{
    /* Dclarations */
    char A[TAILLE_MAX][TAILLE_MAX_CHAINE]; /* tableau donn */
    int N; /* dimension */
    int I; /* rang partir duquel A est tri */
    int J; /* indice courant */
    int K;
    char SWAP[TAILLE_MAX_CHAINE]; /* pour la permutation */
    int FIN; /* position o la dernire permutation a eu lieu. */
    /* permet de ne pas trier un sous-ensemble dj tri. */
    /* Saisie des donnees */
    printf("Dimension du tableau (max.%d) : ",TAILLE_MAX);
    scanf("%d", &N );
    for (J=0; J<N; J++)
    {
        printf("Element %d : ", J);
        scanf("%s", A[J]);
    }
    /* Affichage du tableau */
    printf("Tableau donn :\n");
    for (J=0; J<N; J++)
        printf("%s ", A[J]);
    printf("\n");
    /* Tri du tableau par propagation de l'element maximal. */
    for (I=N-1 ; I>0 ; I=FIN)
    {
        FIN=0;
        for (J=0; J<I; J++)
        {
            K=0;
            /* ORDRE LEXICOGRAPHIQUE RECHERCHE DU PREMIER ELEMENT DISTINCT */
            while( (A[J][K]==A[J+1][K]) && (A[J][K]!='\0') && (A[J+1][K]!='\0'))
            {
                K++;
            }
            if (A[J][K]>A[J+1][K])
            {
                FIN=J;
                /* COPIE DES CHAINE DE CARACTERES*/
                for (K=0;K<TAILLE_MAX_CHAINE;K++)
                {
                    SWAP[K]=A[J][K];
                    A[J][K]=A[J+1][K];
                    A[J+1][K]=SWAP[K];
                }
            }
        }
    }
    /* Edition du resultat */
    printf("Tableau tri :\n");
    for (J=0; J<N; J++)
        printf("%s ", A[J]);
    printf("\n");
    return 0;
}
```