1. Exercice 1 (7 point)

a) Qu'affiche le programme ?

```
void main()
{      int n1,n2,*p1,*p2 ;

       n1 = 8 ;
       p1= &n2 ;
       *p1 = n1 % 5 ;
       p2 = p1 ;
       printf("\n n1= %d n2= %d, p1 = %d  p2= %d", n1,n2,*p1,*p2) ;
       n1*=n2+3 ;
       p2= &n1 ;
       n2++ ;
       printf("n1= %d n2= %d, p1 = %d  p2= %d", n1,n2,*p1,*p2) ;
       n1-- ;
       if (*p2%2)
              printf("\n partie alors ") ;
       else
              printf("\n partie sinon ") ;
}
```

```
n1= 8     n2= 3,     p1 = 3  p2= 3
n1= 48    n2= 4,     p1 = 4  p2= 48
partie alors                        (*p2 vaut 47, donc 47%2 vaut 1)
```

b) Un petit programme

Ecrire un programme qui calcule la réduction offerte à un passager en fonction de son âge. Le programme demande le prix du billet et l'âge du passager, calcule la réduction et affiche le nouveau tarif en fonction de la grille de tarif suivante :

Age ≥ 60 ans	40% de réduction
Age compris entre 26 et 59 ans	pas de réduction
Age compris entre 16 et 25 ans	25 % de réduction
Age compris entre 4 et 15 ans	50% de réduction
Age < 4 ans	gratuit

Le programme demandera à l'utilisateur de ressaisir l'âge du passager et le montant, si ceux-ci sont négatifs.

```
void main()
{
       float prix; /* prix du bille : donnee modifiee t*/
       int age;     /* age du passager  : donnee */

       printf("Calcul du nouveau tarif  appliquee en fonction de l'age et du prix initial
\n") ;
       do {
              printf(" donner le prix initial svp:");
              scanf("%f", &prix);
              printf(" donner votre age svp:");
              scanf("%d", &age);
       }
       while (age<0 || prix<0);

       if (age >= 60 ) prix *= (1.-0.4);                          /* 40% de réduction */
       else
              if (age >= 16 && age <=25) prix *= (1.-0.25); /* 25 % de réduction */
              else
                     if (age >= 4 && age <=15) prix *= (1.-0.5);   /* 50 % de réduction */
                     else
                            if (age < 4) prix = 0.;          /* gratuit */
       printf(" \n le nouveau tarif est %f\n ",prix );
}
```

a) }Deux petites fonctions

Ecrire une fonction *decimale1(x)* qui retourne 1 si le paramètre réel x a une partie décimale non nulle et 0 sinon.
Exemple : *decimale1(4.5)* retourne 1 et *decimale1(4.0)* retourne 0

Ecrire une fonction *decimale2(x)* qui retourne la partie décimale de son paramètre réel x.
Exemple : *decimale2(4.5)* retourne 0.5 et *decimale2(4.0)* retourne 0.0

```
int decimale1(float x)
{
	int xtoint; /* utilisé pour la conversion de x en entier*/

	xtoint= (int) x;
	if (( x-(float)xtoint) !=0.)
		return 1;
	else return 0;
}

float decimale2(float x)
{
	int xtoint; /* utilisé pour la conversion de x en entier*/
	float dec;  /* resultat */

	xtoint= (int) x;
	dec =  x-(float)xtoint;
	if (dec<0)
		dec = -dec;
	return dec;
}
```

2. Code-Barres (6 points) (Copie 2)

On souhaite calculer le « check digit » d'un code produit afin de composer un code barre de type EAN (European Article Number). Le code produit est composé de n-1 chiffres et on souhaite rajouter un dernier chiffre appelé « check digit » qui permettra aux systèmes de lecture de vérifier la cohérence du code barre composé de n chiffres.
La méthode est la suivante :
Tout d'abord on calcule la somme S des chiffres multipliés alternativement par 1 ou 3 (on commence par le chiffre des unités que l'on multiplie par 3 puis le chiffre des dizaines est multiplié par 1, etc.…). Le « check digit » est alors égal à 10 – (le reste de la division entière de S par 10).
Le code barre est ensuite composé du code produit auquel on rajoute un chiffre, le « check digit », par la droite (chiffre des unités).
La méthode est illustrée à l'aide de l'exemple ci-dessous :
Soit le code produit 761234567891.

S = (1*3) + (9*1) + (8*3) + (7*1) + (6*3) + (5*1) + (4*3) + (3*1) + (2*3) + (1*1) + (6*3) + (7*1)
= 3 + 9 + 24 + 7 + 18 + 5 + 12 + 3 + 6 +1 + 18 + 7
= 113

Le reste de la division entière de 113 par 10 est égal à 3. Le « check digit » est alors égal à 10 – 3 = 7
Le code barre est donc 7612345678917
Ecrire le programme qui demande à l'utilisateur le code produit (sous forme d'un entier) et **calcule** le code barre correspondant avant de l'afficher. Cette action sera répétée jusqu'à ce l'utilisateur indique qu'il veut sortir du programme.
Remarques importantes :

1) nous supposerons dans cet exercice que le type int du langage C accepte les entiers longs.
2) Le nombre de chiffres d'un code produit n'est pas déterminé a priori.

```
void main()
{
   int  codeP;              /*  code produit: donnee à ne pas modifier */
   int mult;                /* multiplicateur du chiffre courant */

   int code;                /*  code produit qui sera modifié */
   int somme ;              /* somme */
   int CD;                  /* check digit */
   int codeB;               /* code Barre */

   char cont;               /* continue */

   printf("Calcul d'un code barre a partir d'un code produit \n") ;
   do {
        printf(" donner le code produit  svp:");
```

```
            scanf("%d", &codeP);

            code= codeP; /* conserve codeP */
            somme =0;    /* necessaire d'initialiser dans la boucle do while */
            mult=3;      /* idem */

            while (code != 0) {
                  somme += code%10*mult;
                  if (mult==1)    /* alternance 3 et 1  */
                        mult=3;
                  else
                        mult=1;
                  code/= 10;
            }

            CD= 9 -somme %10;  /* attention, erreur d'enonce il faut soustraire à 9 et
   non à 10 */
            codeB= codeP*10+CD;
            printf("le code barre est %d", codeB) ;

            printf(" voulez vous continuer? [o/n]");
            fflush(stdin);                  /* non demandé */
            cont=getchar();
      }
      while (cont=='o');
   }
```

---

3. Nombres magiques (7 points) (Copie 3)

On veut afficher la figure suivante :

```
*********************************
*         9 * 9 + 7 = 88        *
*        98 * 9 + 6 = 888       *
*       987 * 9 + 5 = 8888      *
*      9876 * 9 + 4 = 88888     *
*     98765 * 9 + 3 = 888888    *
*    987654 * 9 + 2 = 8888888   *
*   9876543 * 9 + 1 = 88888888  *
*  98765432 * 9 + 0 = 888888888 *
*********************************
```

Ecrire le programme permettant d'afficher la figure précédente. Le programme n'utilisera que des appels à la fonction *putchar()*, il n'y aura aucun appel à *printf()*.

Remarque importante : Vous pouvez utiliser la fonction supposée connue *char itoc(int)* qui transforme l'entier passé en paramètre (compris entre 0 et 9) en caractère. Exemples *itoc(8)* retourne '8', *itoc(0)* retourne '0'.

```
void main()
{
  int i,j;

  /* ligne d'entete */
  putchar('\n');
  for (i=1;i<=32;i++)
       putchar('*');
  /* ligne 1 à 8 */
  for (j=1; j<=8; j++) {
       putchar('\n');
       putchar('*');
       for(i=1; i<=9-j;i++)
             putchar(' ');
       for(i=9 ; i> 9-j; i--)
             putchar(itoc(i));
       putchar(' '); putchar('*'); putchar(' ');putchar('9'); putchar(' '); putchar('+');
       putchar(' '); putchar(itoc(9-j-1)); putchar(' ');putchar('=');putchar(' ');
       for(i=1; i<=j+1; i++)
             putchar('8');
       for(i=1; i<=9-j;i++)
             putchar(' ');
       putchar('*');
       }
  /* derniere ligne */
  putchar('\n');
  for (i=1;i<=32;i++)
       putchar('*');
}
```