

Exercice 3 série 4:

```
typedef struct liste { short int partie;
    struct liste* suiv;
} ListeNombre;
```

```
ListeNombre* ajouterEnFin(ListeNombre* TeteActuelle, short int n)
{ ListeNombre* nouvel = (ListeNombre*)malloc(sizeof(ListeNombre));
  nouvel->partie = n;
  nouvel->suiv = NULL;
```

```
    if(TeteActuelle == NULL)
    { return nouvel; }
    else
    { ListeNombre* temp=TeteActuelle;
      while(temp->suiv != NULL)
      { temp = temp->suiv; }
      temp->suiv = nouvel;
      return TeteActuelle;
    }
}
```

```
void initialiserZero(short int c[4])
{int i;
  for(i=0;i<4;i++) c[i]=-1;
}
```

```
void cycler(char c[4], int k)
{int i,j; for(i=0;i<k;i++) {for(j=3;j>=1;j--) c[j]=c[j-1]; c[0]=0; } }
short int ConvertToShort(char c[4])
{int i,j,k=0; short int m;
  for(i=0;i<4;i++) if(c[i]==-1)k++;
  cycler(c,k);
  m=c[3]+10*c[2]+100*c[1]+1000*c[0]; return(m); }
```

```
ListeNombre* RegrouperChiffres(char* S)
{ListeNombre* Tete=NULL;
  int i=0,j=0,k;
  short int c[4];
  while(S[i]!='\0')
  {initialiserZero(c);
    for(j=0;j<4 && S[i]!='\0'; j++)
    {c[j]=S[i]-48; i++;}
    k=ConvertToShort(c);
    Tete=ajouterEnFin(Tete,k);
  }
  return (Tete);
}
```
