TD 2 : boucles et tableaux

Programmation en C (LC4)

Semaine du 5 février 2007

1 Boucles, tableaux à une ou deux dimensions

▶ **Exercice 1**

```
int somme_for(int n, int tab[]){
  int i, tmp = 0;
  for(i = 0; i < n; i++)
    tmp += tab[i];
  return tmp;
}

int somme_while(int n, int tab[]){
  int i, tmp = 0;
  while(i < n){
    tmp += tab[i];
    i++;
  }
  return tmp;
}

int somme_do_while(int n, int tab[]){
  int i, tmp = 0;
  do{
    tmp += tab[i];
    i++;
  } while(i < n);
  return tmp;
}
```

▶ **Exercice 2**

```
int inverse_nb(int nb){
  if(nb<=0 || nb>=10000){
    printf("le nombre n'est pas compris entre 0 et 10000");
    return 0;
    }
  while(nb > 0){
    printf("%i",nb%10);
    nb = nb/10;
  }
  return 1;
}
```

▶ **Exercice 3**

```
void tasse(int n, int tab[]){
  int i, j = 0;
  for(i = 0; i < n; i++){
    tab[j] = tab[i];
    if(tab[i] != 0)
        j++;
  }

  for(; j < n; j++)
    tab[j] = 0;
}
```

► **Exercice 4**

```
void inverse(int n,int tab[]){
  int i, tmp = 0;
  for(i = 0; i < n/2; i++){
    tmp = tab[i];
    tab[i] = tab[n-1-i];
    tab[n-1-i] = tmp;
  }
}
```

► **Exercice 5**

```
void fusion(int n, int tabA[], int m, int tabB[], int fus[]){
  int a = 0;
  int b = 0;
  /* On remplit d'abord fus[] jusqu'a ce qu'on arrive a la fin de tabA[] ou de tabB[] */
  while(a<n && b<m) {
     if(tabA[a] < tabB[b]){
               fus[a+b] = tabA[a];
               a++;
     }
     else{
               fus[a+b] = tabB[b];
               b++;
     }
  }
  /* On remplit ensuite fus[] par ce qui reste */
  while(a+b < n+m){
    if(a >= n){
      fus[a+b] = tabB[b];
      b++;
    }
    else{
      fus[a+b] = tabA[a];
      a++;
    }
  }
}
```

► **Exercice 6**

```
void matrice_ligne(int i, int j, int matrice[][j],int ligne[]){
  int x,y;
  for(y = 0; y < i; y++)
```

```
    for(x = 0; x < j; x++)
      ligne[x + y*j] = matrice[y][x];
}
```

▶ **Exercice 7**

```
void insere(int n, int tab[], int val){
  int tmp, i = 0;
  for(i = 0; i < (n-1); i++){
    if(tab[i] > val){
      tmp = tab[i];
      tab[i] = val;
      val = tmp;
    }
  }
  tab[n-1] = val;
}
```

▶ **Exercice 8**

```
int factorielle_boucle(int n){
  int i, tmp = 1;
  for(i = 1; i <= n; i++)
    tmp = tmp*i;
  return tmp;
}


int factorielle_rec(int n){
  if (n == 0)
      return 1;
  else
      return(n * factorielle_rec(n-1) );
}
```

2 Reprise du TP1

▶ **Exercice 9**

```
int couleur_en_entier(int rouge, int vert, int bleu){
  return ((rouge << 16) | (vert << 8) | bleu);
}
```

▶ **Exercice 10**

```
int rouge( int code){
  return (code >> 16);
}

int vert(int code){
  return (code & 0xff00) >> 8;
}

int bleu(int code){
  return (code & 0xff);
}
```