

TD6 : Révisions

Programmation en C (LC4)

Semaine du 2 mars 2008

1 Types

Exercice 1 Qu'affiche la seconde ligne du programme suivant, en sachant que la première ligne affichée est : « `prem : 61, sec : 72, tri : 9, qua : 20` »

```
void blabla(void){
    char prem='a';
    char sec='r';
    char tri='\t';
    char qua='_';
    unsigned int res;
    printf("prem:_%x,_sec:_%x,_tri:_%x,_qua:_%x\n",
           prem, sec, tri, qua );
    res = (qua << 24) | (tri << 16) | (sec << 8) | prem;
    printf("res:_%x\n",res);
}
```

Remarque : `%x` permet d'afficher les données en hexadécimal.

Exercice 2 Les caractères constituant les lettres de l'alphabets sont codées, dans le code ASCII, les uns après les autres, c'est à dire que la valeur entière de 'b' est celle de 'a'+1. Ecrivez une fonction « `void majuscule(char* mot)` » qui prend un mot en minuscule (on ne demande pas de tester si le mot est bien en minuscules) et qui le modifie pour qu'il soit en majuscule. Ecrire l'appel à cette fonction dans le `main` pour qu'elle modifie un mot passé en argument.

► **Correction**

```
void majuscule(char* mot){
    while(*mot != '\0'){
        (*mot) = (*mot) + 'A' - 'a';
        mot++;
    }
}

int main(int argc, char ** argv){
    majuscule(argv[1]);
    printf("%s\n",argv[1]);
}
```

Exercice 3 Qu'affiche la fonction suivante ?

```
void melange(void){
    int i=36;
    float j=6.2;
    printf("%i,_%i\n", i/((int)j), (int)(i/j));
}
```

2 Tableaux

Exercice 4 On suppose qu'on manipule des tableaux de `double`. Ecrivez les fonctions qui font les actions suivantes :

- renvoyer le plus grand élément d'un tableau
- inverser l'élément `i` avec l'élément `j`
- renverser l'ordre des éléments d'un tableau
- vérifier si un tableau est trié dans l'ordre croissant
- prendre en argument deux tableaux et renvoyer la concaténation de ces deux tableaux

► **Correction**

```
double max(int n, double t[]){
    double res=t[0];
    int i=0;
    for(;i<n;i++){
        if(t[i]>res)
            res = t[i];
    }
    return res;
}

void swap(int i, int j, int n, double t[]){
    double tmp=t[i];
    t[i]=t[j];
    t[j]=tmp;
}

void inv(int n, double t[]){
    double tmp;
    int i=0;
    for(;i<n/2;i++){
        swap(i,n-i-1,n,t);
    }
}

int verif(int n, double t[]){
    int i=0;
    for(;i<n-1;i++){
        if(t[i]>t[i+1])
            return 0;
    }
    return 1;
}

double* concat(int n,double t1[], int m, double t2[]){
    double *res=malloc(sizeof(double)*(n+m));
    int i=0;
    for(;i<n;i++){
        res[i]=t1[i];
    }
    for(i=0;i<m;i++){
        res[i+n]=t2[i];
    }
    return res;
}
```

3 Chaînes de caractères

Exercice 5 Ecrivez les fonctions qui ont les propriétés suivantes :

- renvoyer la longueur d'une chaîne de caractères
- prendre une chaîne de caractères `phrase` en entrée et renvoyer un tableau de chaînes de caractères des mots de cette phrase

- tester si une chaîne de caractère **mot** a une sous-chaîne continue de caractère **partie**. Par exemple **zazbcrt** contient **zaz**.
- tester si une chaîne de caractères **mot** a une sous-chaîne possiblement éclatée. Par exemple **zazbcrt** contient **abc**.

► **Correction**

```
int longueur(char* m){  
    int res=0;  
    while(*m != '\0'){  
        res++;  
        m++;  
    }  
    return res;  
}
```