ISEN école d'ingénieurs

TP N°7

Les chaines de caractères

Préambule

En langage C il est possible de passer des paramètres à un programme lors de son exécution. Par exemple pour un programme multipliant deux nombres on pourrait avoir :

```
> gcc Main.c –o Preambule
> ./Preambule 5 4
Le résultat de 5*4 est 20.
```

Pour cela on utilise les une fonction main différente de d'habitude :

```
int main (int argc, char** argv);
```

Ici le paramètre *argc* donne le nombre de paramètre donné au programme (Attention le nom du programme est considéré comme un paramètre). Dans le cas précédent *argc* à pour valeur 3. Le paramètre *argv* correspond à un tableau de chaine de caractère contenant chaque paramètre sous formes de chaine. Ici on a donc :

- argv[0] qui vaut "./Tp5_Exo1"
- argv[1] qui vaut "5" (ici 5 n'est pas un entier mais une chaine de caractère contenant 5). La fonction *atoi ()* permet de convertir une chaine de caractère en entier.
- argv[2] qui vaut "4" (ici 4 n'est pas un entier mais une chaine de caractère contenant 4). La fonction *atoi ()* permet de convertir une chaine de caractère en entier.

Le programme précédent s'écrierait en langage C comme suit :

```c
// Includes.
#include <stdio.h>
#include <stdlib.h>

// Point d'entree du programme.
int main (int argc, char** argv)
{
	int nombre1;
	int nombre2;

	// On test les paramètres du programme.
	if (argc != 3)
	{
		printf ("L'utilisation du programme est incorrect !!\n");
		printf ("Utilisation :\n");
		printf ("\t%s nombre1 nombre2\n\n", argv[0]);

		return EXIT_FAILURE;
	}

	// On récupère les deux entiers.
	nombre1 = atoi (argv[1]);
	nombre2 = atoi (argv[2]);
```

```
        // On affiche le résultat.
        printf ("Le résultat de %d*%d est %d.\n", nombre1, nombre2,
                nombre1*nombre2);

        return EXIT_SUCCESS;
}
```

1. Nombre de voyelles

Ecrivez un programme principal permettant de calculer le nombre de voyelles contenues dans la chaine de caractère passée en paramètre du programme par l'utilisateur. Le comportement de votre programme devra être le suivant :

```
> ./Tp5_Exo1 bonjour
Le nombre de voyelles de bonjour est 3.
```

2. Tableau de chaines

Ecrivez un programme principal permettant de demander à l'utilisateur un certain nombre de chaines (au maximum 10), de les stocker dans un tableau et de les afficher dans l'ordre inverse de leur saisie. Le nombre de chaines sera un paramètre du programme. On pourra écrire une fonction *SaisieChaine ()* pour saisir une chaine. Le comportement de votre programme devra être le suivant :

```
> ./Tp5_Exo2 3
1. Entrez une chaine > beau
2. Entrez une chaine > fait
3. Entrez une chaine > il
La phrase est : il fait beau.
```

3. Passage en majuscules

Ecrivez une fonction *Majuscules* qui permet de mettre en majuscules une chaine de caractère entré par l'utilisateur au clavier. On utilisera le mécanisme vu précédemment pour permettre à l'utilisateur de rentrer une chaine comme paramètre du programme. Si ce n'est pas le cas on lui demandera de rentrer une chaine comme sans l'exemple ci-dessous. Le comportement de votre programme devra être le suivant :

```
> ./Tp5_Exo3
Entrez une chaine : bonjour
Votre chaine en majuscules est : BONJOUR

> ./Tp5_Exo3 salut
Votre chaine en majuscules est : SALUT
```

4. Palindrome

Ecrivez une fonction *Palindrome* qui permet de savoir si une chaine est un palindrome (la fonction renvoie 1) ou non (la fonction renvoie 0). Un palindrome est une figure de style désignant un mot/phrase qui peut se lire, de façon identique, de gauche à droite ou de droite à gauche. Le comportement de votre programme devra être le suivant :

```
> ./Tp5_Exo4
Entrez une chaine : bonjour
"bonjour" n'est pas un palindrome.

> ./Tp5_Exo4 kayak
"kayak" est un palindrome.
```

5. Anagramme

Ecrivez une fonction *Anagramme* qui permet de savoir si deux chaines sont l'anagramme l'une de l'autre (la fonction renvoie 1) ou non (la fonction renvoie 0). Une anagramme est une figure de style désignant deux mots/phrases qui contiennent exactement les mêmes lettres dans un ordre différent. Le comportement de votre programme devra être le suivant :

```
> ./Tp5_Exo5
Entrez une chaine : lampe
Entrez une chaine : palme
"lampe" et "palme" sont des anagrammes.

> ./Tp5_Exo5 bonjour salut
"bonjour" et "salut" ne sont pas des anagrammes.
```