COMPILATION TP 3
EMSI - 4ÈME IIR
2017/2018

Prof. M. D. RAHMANI

Rappel: traitement de caractères

Les macros d'analyse de caractères:

Ces macros sont incluses dans la bibliothèque `<ctype.h>`.

Elles acceptent comme argument un `char` ou un `int` et retournent un entier différent de 0 si l'argument est compris dans les limites indiquées ci-dessous:

macros	condition
`isalpha(c)`	A-Z, a-z
`isupper(c)`	A-Z
`islower(c)`	a-z
`isdigit(c)`	0-9
`isxdigit(c)`	0-9, A-F, a-f
`isspace(c)`	blanc
`isalnum(c)`	0-9, A-Z, a-z

TP Implantation d'un analyseur lexical

- **Exercice 1:** Ecrire un analyseur lexical qui reconnait les unités lexicales suivantes :
  - les nombres entiers réels,
  - les identificateurs,
  - les opérateurs relationnels,
  - les parenthèses,
  - Votre analyseur doit retourner et afficher les lexèmes et les unités lexicales correspondantes.

Remarques :

- Les espaces et les tabulations doivent être ignorés.
- L'analyseur lexical doit retourner un code d'erreur pout tout autre caractère inconnu.
- Il est préférable de commencer par l'entrée du texte à analyser par l'entrée standard et d'afficher le résultat à l'écran de sortie standard.

TP Implantation d'un analyseur lexical

Exercice 1:

Indications : les prototypes

```
int delim(char *chaine, int i);
int identificateur(char *chaine, int i);
int reel(char *chaine, int i);
int operateur (char *chaine, int i);
int parenthese (char *chaine, int i);
void analyse(char *chaine);
```

La fonction principale: `main`

```
int main(int argc, char **argv)
{
    char chaineATraiter[100];
    printf(" Donnez une chaine a analyser : \n\t");
    gets(chaineATraiter);
    analyse(chaineATraiter);

    return 0;
}
```

Exercice 1 :

Solution : **exo1.cpp**

Exercice 2 :

Solution : **exo2.cpp**