Td No 2

Exercice 1

On considère le programme suivant:

```
#include <stdio.h>
#include <stdlib.h>
int main(void){
  float r;
  r=2;
  printf("p(%f)=%f\ns(%f)=%f\n",r,p(r),r,s(r));
  return EXIT_SUCCESS;
}
float c(float e){
  return e*e;
}
float p(float x){
  return 2.0*pi()*x;
}
float s(float x){
  float res;
  res=pi()*c(x);
  return res;
}
int pi(void){
  return 3.14159265358;
}
```

On vous demande :

- d'y détecter les erreurs qui s'y trouvent
- d'indiquer ce qu'il fait
- de le taper dans l'environnement Rhide et de l'exécuter (dont une fois pas à pas)

Exercice 2 Bonjour

On souhaite écrire une fonction nommée bonjour qui affiche à l'écran un court message de bienvenue:

- quel sera le prototype de votre fonction ?
- Ecrire la fonction ainsi qu'un court programme qui l'utilise

2004-2005 modélisation en C Deug SDM 2

Exercice 3 max

Question 1 affichage du max

Ecrire une fonction qui, à partir de deux valeurs qui lui sont fournies, affiche à l'écran la plus grande.

Question 2 calcul du max

En pratique, on dissocie les fonctions qui font des calculs dont le résultat est utilisable ailleurs des fonctions qui réalisent des affichages. Ecrire une fonction qui, à partir de deux valeurs qui lui sont fournies, **retourne** la plus grande. Vous écrirez un court programme qui l'utilise.

Question 3 calcul du max de 3 nombres

Ecrire une fonction qui, à partir de trois valeurs qui lui sont fournies, **retourne** la plus grande. Vous écrirez un court programme qui l'utilise.

Exercice 4 équitation du second degré

Question 1 nombre de solutions réelles

Ecrire une fonction qui, à partir de trois valeurs a, b et c retourne le nombre de solutions de l'équation $ax^2+bx+c=0$.

Question 2 solutions

Ecrire deux fonctions sol1 et sol2 qui retournent respectivement les deux solutions d'une équation du second degré

Question 3 assert

Modifiez les fonctions écrites à la question précédente de façon à refuser de fournir des solutions quand il n'y en a pas.

Pour cela, on utilisera la fonction assert (fournie par assert.h) qui a le comportement suivant: assert (condition):

- si la condition vaut 0, le programme continue son exécution
- si la condition a une valeur différente de 0, l'exécution du programme est interrompue.

assert est un outil minimal utile pour préciser le domaine de définition des fonctions et imposer des conditions vitales au bon fonctionnement du programme.

Exercice 5 puissance

Question 1 estPair

Ecrire une fonction qui indique si un entier *n* est pair. On rappelle que le reste de la division de a par b se note a%b. On pourra s'intéresser au reste de la division de *n* par 2.

Question 2 puissance version 1

On peut définir x^n par $x^n=x*x^{n-1}$ si n >0 et $x^0=1$. Ecrire une fonction qui calcule x^n en utilisant cette

Pascal PETIT pascal.petit@lami.univ-evry.fr

définition. Une telle fonction qui contient des appels vers elle-même est appelée une fonction récursive.

Combien d'appel à la fonction puissance nécessite le calcul de 2^7, de 2^{10}.

Question 3 puissance version 2

On se propose d'utiliser la définition équivalente suivante : x^n par $x^n=x*x^{n-1}$ si n est impair; $x^n=x^{n-1} * x^{n-1}$ si n est pairet $x^0=1$