Examen partiel - semestre 1
Initiation à la programmation 2014/2015

Électronique et Physique appliquée
Matériaux et chimie
Correction

Tous documents
non électroniques autorisés
durée 1h

Nom, prénom : Spécialité :

- **Répondre sur le document.**
- Écrire vos programmes en langage C en respectant la syntaxe et l'indentation usuelle ;
- Soignez votre présentation ;
- Les questions sont relativement indépendantes ;
- Chaque question est notée sur 3 points.

1. Analyse de code

Qu'affiche le programme ci-dessous ?

```
#include<stdio.h>
int main (void) {
    int a = 1/3 ;
    float b = 1/3 ;
    float c = 1/3.0 ;
    int d = 22 % 12 ;
    int e = ( ( 7>3 ) && ( 8>2) ) ;
    int f = ( 9 != ( 3*3 ) ) ;
    char g='g'+2;
    printf ("a = %d, b = %f, c = %f\n", a, b, c) ;
    c += 0.000000001 ;
    printf ("c = %f, d = %d, d = %x, e = %d, f = %d, g = %c\n", c, d, d, e, f, g) ;
}
```

```
a = 0, b = 0.000000, c = 0.333333
c = 0.333333, d = 10, d = a, e = 1, f = 0, g = i
```

2. Analyse de code

Qu'affiche le programme ci-dessous et pourquoi ?

```
#include<stdio.h>
int main (void) {
          float s1 = 1, q = 0.5, r = 0.5, s2 ;
          do {
             s2 = s1 ;
             s1 = s1 + r ;
             r = r * q;
             if (s1 == s2 ) break ;
          } while ( 1 ) ;
          printf ("somme = %f\n", s1 ) ;
}
```

```
somme = 2.000000
C'est la somme de 1 + 1/2 + 1/4 + 1/8 + 1/16.... = 1/(1-q) = 2
```

3. Programmation : boucles

Écrire un programme qui affiche les valeurs du sinus de tous les angles de 0 à 90 degrés, par pas de 15 degrés. Vous utiliserez la fonction *sin(x)* qui retourne le sinus de *x* si *x* est en radian. *x* doit également être de type *double*. On souhaite un affichage de la forme :

```
sin(0) = 0.000000
sin(15) = 0.258819
sin(30) = 0.500000
...
sin(90) = 1.000000
```

```
#include<stdio.h>
#include<math.h>
int main (void) {
    int i ;
    for (i=0 ; i <= 90 ; i=i+15)
        printf ("sin(%d) = %lf\n",i, sin(i/180.0*3.1415926) ) ;
}
```

4. Programmation : tests

Écrire un programme qui demande des nombres entiers positifs à l'utilisateur. Lorsque l'utilisateur entre un nombre négatif, le programme s'arrête et affiche le minimum, le maximum et la moyenne de la série. Vous ne traiterez pas le cas où l'utilisateur commence par saisir un nombre négatif. Voici une trace d'exécution :

```
Entrez un nombre entier positif : 5
Entrez un nombre entier positif : 1
Entrez un nombre entier positif : 0
Entrez un nombre entier positif : 6
Entrez un nombre entier positif : -1
min = 0, max = 6, moy = 3.000000
```

```
#include<stdio.h>
int main (void) {
     int n, min, max, som , i ;
     printf ("Entrez un nombre entier positif : ") ;
     scanf ("%d", &n) ;
     min = max = som = n ;
```

```
      i = 1 ;
      for(;;) {
            printf ("Entrez un nombre entier positif : ") ;
            scanf ("%d", &n) ;
            if (n < 0) break ;
            if (n < min) min = n ;
            if (n > max) max = n ;
            som = som + n ;
            i = i+1 ;
      }
      printf ("min = %d, max = %d, moy = %f \n", min, max, som/(float)i) ;
}
```

5. Programmation : tests

Écrire de nouveau le programme précédent avec la contrainte que l'utilisateur peut maintenant saisir un nombre négatif dès le premier nombre. Voici une trace d'exécution :

```
Entrez un nombre entier positif : -4
Pas de nombre saisi
```

```
#include<stdio.h>
#include<stdlib.h>
int main (void) {
      int n, min, max, som , i ;
      printf ("Entrez un nombre entier positif : ") ;
      scanf ("%d", &n) ;
      if (n <0 ) {
            printf ("Pas de nombres entrés") ;
            exit (0) ;
      }
      min = max = som = n ;
      i = 1 ;
      for(;;) {
            printf ("Entrez un nombre entier positif : ") ;
            scanf ("%d", &n) ;
            if (n < 0) break ;
            if (n < min) min = n ;
            if (n > max) max = n ;
            som = som + n ;
            i = i+1 ;
      }
      printf ("min = %d, max = %d, moy = %f \n", min, max, som/(float)i) ;
}
```

6. Programmation : carrés

Écrire un programme qui demande un nombre entier N, supérieur à 2, à l'utilisateur. Puis le programme affiche un carré de N caractères de côté. Voici deux traces d'exécution :

```
Entrez un nombre entier > 2 : 4
4444
4  4
4  4
4444
```

```
Entrez un nombre entier > 2 : 3
333
3 3
333
```

```
#include<stdio.h>
int main (void) {
      int n, i, j ;
      printf ("Entrez un nombre entier > 2 : ") ;
      scanf ("%d", &n) ;
      for (i=0 ; i< n ; i++ )
            printf ("%d",n) ;
      printf ("\n") ;
      for (i=0 ; i< n-2 ; i++ ) {
            printf ("%d", n) ;
            for (j=0 ; j< n-2 ; j++ )
                  printf (" ") ;
            printf ("%d\n",n) ;
      }
      for (i=0 ; i< n ; i++ )
            printf ("%d", n) ;
      printf ("\n") ;
}
```

7. Programmation : fonctions

Écrire la fonction *distance* qui retourne la distance entre 2 points. La fonction prendra en paramètre les coordonnées x1, y1, x2, y2 des 2 points et retournera la distance. La fonction *sqrt(x)* retourne la racine carrée de *x* si le type de *x* est *double*.

Voici le *main* permettant d'utiliser la fonction :

```
#include<stdio.h>
#include<math.h>

int main (void) {
          printf ("d = %f  \n", distance (1.0, 1.0, 4.0, 5.0 )) ;
}
```

Et une trace d'exécution :

```
d = 5.000000
```

```
float distance (float x1, float y1, float x2, float y2 ) {
      float d1 = x2-x1 ;
      float d2 = y2-y1 ;
      return (sqrt((d1*d1)+(d2*d2)) );
}
```