

exercice 2:

Programmation Série 2

écrire un programme C qui lit trois variables réelles a, b, c ensuite le programme :

1. effectue une permutation circulaire de ces variables (transfère à a la valeur de b , à b la valeur de c et à c la valeur de a) et affiche les valeurs des variables avant et après permutation
2. affiche la plus grande des trois valeurs.

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

```
main()
```

```
{ float a=0.0, b=0.0, c=0.0, temp1=0.0, temp2=0.0, max=0.0;
```

```
  printf("Entrez la valeur de trois variables a, b et c \n");
```

```
  scanf("%f%f%f", &a, &b, &c);
```

```
  printf("Les valeurs entrées avant permutation : \n");
```

```
  printf(" a = %f, b = %f, c = %f", a, b, c);
```

```
  temp1 = a;
```

sans utiliser temp2 :

```
  a = b;
```

```
  temp1 = a;
```

```
  temp2 = c;
```

```
  a = b;
```

```
  c = temp1;
```

```
  b = c;
```

```
  b = temp2;
```

```
  c = temp1;
```

```
  printf("après permutation : \n");
```

```
  printf(" a = %f, b = %f, c = %f", a, b, c);
```

autre méthode

```
  if (a > b) {
```

```
    max = a;
```

```
    if (a > c) { max = a; } else { max = c; } }
```

```
  if (b > max)
```

```
    max = b;
```

```
  else {
```

```
  if (c > max)
```

```
    max = c;
```

```
    if (b > c) { max = b; } else { max = c; } }
```

```
  printf("La plus grande des trois valeurs est : %f \n", max);
```

```
}
```

exercice 1:

écrire un programme qui affiche le nombre d'octets occupés par les types :

char (1 octet)	⇐ entiers	float (4 oc)	⇐ réel
short (2 oc)		double (8 oc)	
int (2 ou 4 oc)		long double (10 oc)	
long (4 oc)			

⇒ voir TP

on utilisera printf et sizeof !

exercice 3 :

a - n = 0 ⇒ Nul
petit

b - n = 1 ⇒ petit

c - n = 4 ⇒ moyen
grand

d - n = 10 ⇒ grand

e - n = -5 ⇒ grand

exercice 1: série 2 programmation

```
#include <stdio.h>
main ()
{
    printf("Le nombre d'octets occupées par le  
type char : %d\n", sizeof(char));
    ...
}
```

22-10-2014

exercice 4:

```
1/  
# include <stdio.h>  
main ()  
{  
    int N, i;  
    float S = 0;  
    printf ("Entrez le nombre N.");  
    scanf ("%d", &N);  
    for (i = 1; i <= N; i++)  
        S += 1.0/i;  
    printf ("La somme vaut %f\n", S);  
}
```

&1 TP

exercice 5:

```
# include <stdio.h>  
main ()  
{  
    int i, N, AVD = 1, D = 1, nouveau;  
    printf ("Entrez N\n");  
    scanf ("%d", &N);  
    if (N == 1 || N == 2) nouveau = 1;  
    else for (i = 3; i <= N; i++)  
        { nouveau = AVD + D;  
          AVD = D;  
          D = nouveau;  
        }  
}
```

exercice 1 :

serie 2 prog

21

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

```
main()
```

```
{ float epsilon; int i = 1; float S = 0;
```

```
do {
```

```
    printf("Entrez epsilon \n");
```

```
    scanf("%f", &epsilon);
```

```
} while (epsilon > 1);
```

```
while (1.0/i >= epsilon) {
```

```
    S = S + (float) 1/i;
```

```
    i += 2;
```

```
} while (1.0/i >= epsilon);
```

```
printf ("U(%d) = %d\n", N, nouveau);
}
```

exercice 6:

$$\sin x = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!}$$

u_k

$$u_k = \frac{-x^2}{2k(2k+1)} u_{k-1}$$

```
#include <stdio.h>
main ()
{ float x, u, s;
  int n, i;
  printf ("Entrez x et n\n");
  scanf ("%f %d", &x, &n);
  s = u = x;
  for (k = 1; k <= n; k++)
  { u* = -(x*x) / ((2*k) * (2*k+1));
    s + = u;
  }

  printf ("Somme (%f) = %f\n", x, s);
}
```

exercice 7

1/

```
#include <stdio.h>

main()
{
    int p, i;
    do {
        printf("Entrez un entier p : \n");
        scanf("%d", &p); } while (p <= 1);
    for (i = 2; i < p; i++)
        if (p % i == 0) break;
    if (i < p)
        printf("%d n'est pas un nombre premier\n", p);
    else printf("%d est premier\n", p);
}
```

```
#include <stdio.h>
main()
{ int nombre = 0, i, j;
  for (i = 2; i < 1000; i++) {
    for (j = 2; j < i; j++)
      if (i % j == 0) break;
    if (j == i) nombre++;
  }
  printf("Le nombre des nombre premiers
entre 1 et 1000 est %d \n", nombre);
```

exercice 8 :

```
#include <stdio.h>
main()
{ int A, nombre, i = 1;
  do {
    printf("Entrez la valeur de A tq: A > 0 \n");
    scanf("%d", &A); } while (A <= 0);
  nombre = A;
  printf("x (0) = %d \n", nombre);
  do {
    if (nombre % 2 == 0)
      nombre = nombre / 2;
    else nombre = 3 * nombre + 1;
    printf("x (%d) = %d \n", i, nombre);
    i++;
  } while (nombre != 1)
  printf("x (%d) = %d \n", i, nombre);
```

28-10-2014

exercice 7-8 : (TP)

exercice 9 :

affichage :

continue : on passe
à l'itération
suivante

0 est pair
3 est multiple de 3
9 est multiple de 3
15 est multiple de 3
20 est multiple de 5

on sort de la
boucle

break

