



PROGRAMMATION PROCEDURALE - Année 2019/2020

CLASSE CYCLE D'INGENIEUR ING1G04

INTRODUCTION AU C :

**TP4**

---

Made by :  
Lilian Naretto

```
1 #include "tp4.h"
2
3 /*!\author Lilian Naretto <narettolil@eisti.eu>
4 \date 25 octobre 2019
5 \file tp4.c
6 \brief tp4
7 \version 0.1 premier jet*/
8
9 float approximation(int n){
10     float i,c,x,y,z;
11     c=0;
12     float rayon=1.0;
13     srand(time(NULL));
14     for ( i = 1; i < n; i++)
15     {
16         x=((float)rand())/((float)(RAND_MAX/rayon));
17         y=((float)rand())/((float)(RAND_MAX/rayon));
18         z=pow(x,2)+pow(y,2);
19         if (z<1)
20         {
21             c=c+1;
22         }
23     }
24     return (4*c)/n;
25 }
26
27 float Madhava(int n){
28     float somme=1.0;
29     float x=1.0;
30     for (int i = 1; i < n; i++)
31     {
32         x/=-3.0;
33         somme+=x/(2*i+1);
34     }
35     return sqrt(12.0)*somme;
36 }
37 }
```

```
39 float wallis(int n){
40     float accum;
41     float accum2=2.0;
42     float res=1;
43     for (accum=1.0 ; accum<=n ; accum=accum+2.0)
44     {
45         res=res*(accum2/accum);
46         if (accum==(accum2+1))
47         {
48             accum2=accum2+2.0;
49             accum=accum-2.0;
50         }
51     }
52     res=res*2;
53     return res;
54 }
55
56 float newton(int n){
57     float accum = 1;
58     float racine =1;
59     for ( accum = 1; accum < n; accum++)
60     {
61         racine = ((2/racine)+racine)/2;
62     }
63     return racine;
64 }
65
66 float halley(int n){
67     float accum = 1;
68     float racine =1;
69     for ( accum = 1; accum < n; accum++)
70     {
71         racine = racine*((pow(racine,2)+6)/((3*pow(racine,2)+2)));
72     }
73     return racine;
74 }
```

```
76 float theon(int n){
77     float accum = 1;
78     float racine = 1;
79     float racine2 = 2;
80     for ( accum = 1; accum < n; accum++)
81     {
82         racine = (racine + 2)/(racine + 1);
83     }
84     return racine;
85 }
86 /*! \fn int main(int argc, char** argv)
87 \param argc nombre d'arguments en entrée
88 \param argv valeur des arguments en entrée
89 \brief sert de menu pour choisir les questions voulues*/
90
91 int main(int argc, char** argv){
92     int n;
93     int question;
94     printf("choisis le numero de ta question : ");
95     scanf("%d",&question);
96     if (question > 6 || question < 0)
97     {
98         printf("mauvais numero\n");
99     }
100     else
101     {
102         switch (question)
103         {
104             case 1:
105                 printf("donne un chiffre : ");
106                 scanf("%d",&n);
107                 printf("pi = %f\n",approximation(n));
108                 return 0;
109                 break;
```

```
110             case 2:
111                 printf("donne un chiffre : ");
112                 scanf("%d",&n);
113                 printf("pi = %f\n",Madhava(n));
114                 return 0;
115                 break;
116             case 3:
117                 printf("donne un chiffre : ");
118                 scanf("%d",&n);
119                 printf("pi = %f\n",wallis(n));
120                 return 0;
121                 break;
122             case 4:
123                 printf("donne un chiffre : ");
124                 scanf("%d",&n);
125                 printf("sqrt(2) = %f\n",newton(n));
126                 return 0;
127                 break;
128             case 5:
129                 printf("donne un chiffre : ");
130                 scanf("%d",&n);
131                 printf("sqrt(2) = %f\n",halley(n));
132                 return 0;
133                 break;
134             case 6:
135                 printf("donne un chiffre : ");
136                 scanf("%d",&n);
137                 printf("sqrt(2) = %f\n",theon(n));
138                 return 0;
139                 break;
140             default:
141                 break;
142         }
143     }
144 }
145 }
```

```
1  #ifndef __TP4_H_
2  #define __TP4_H_
3
4  #include <stdio.h> /*Autorise l'emploi de printf et de scanf.*/
5  #include <time.h>
6  #include <stdlib.h>
7  #include <math.h>
8
9  /*! float approximation(int n)
10 \param int n
11 \brief donne une approximation de pi à l'aide de la methode de quadrillage*/
12 float approximation(int n);
13
14 /*! float Madhava(int n)
15 \param int n
16 \brief donne une approximation de pi à l'aide de la methode de Madhava*/
17 float Madhava(int n);
18
19 /*! float wallis(int n)
20 \param int n
21 \brief donne une approximation de pi à l'aide de la methode de jhon wallis*/
22 float wallis(int n);
23
24 /*! float float newton(int n)
25 \param int n
26 \brief donne une approximation de sqrt(2) à l'aide de la methode de newton*/
27 float newton(int n);
28
29 /*! float float newton(int n)
30 \param int n
31 \brief donne une approximation de sqrt(2) à l'aide de la methode de edmund halley*/
32 float halley(int n);
33
34 /*! float float newton(int n)
35 \param int n
36 \brief donne une approximation de sqrt(2) à l'aide de la methode de theon de smyrne*/
37 float theon(int n);
```