



PROGRAMMATION PROCEDURALE - Année 2019/2020

CLASSE CYCLE D'INGENIEUR ING1G04

INTRODUCTION AU C :

TP3

Made by :
Lilian Naretto

1 Question 1

```
1 #include <stdio.h> /*Autorise l'emploi de printf et de scanf.*/
2 #include <stdlib.h> /*Autorise l'emploi de exit.*/
3
4 /*! \author Lilian Naretto <narettolil@eisti.eu>
5 \date 15 octobre 2019
6 \file tp3q1.c
7 \brief question 1 du tp3
8 \version 0.1 premier jet*/
9
10 /*! \fn int saisieentier()
11 \return n, un chiffre
12 \brief demande de saisir un chiffre, le renvoie*/
13
14 int saisieentier(){
15     int n;
16     printf("ecrit un entier : ");
17     scanf("%d",&n);
18     if (n == 0)
19     {
20         exit (1);
21     }
22
23     return n;
24 }
25
26 /*! \fn int main(int argc, char** argv)
27 \param argc nombre d'arguments en entrée
28 \param argv valeur des arguments en entrée
29 \return 0 si tout c'est bien passé
30 \brief demande de saisir un chiffre, le renvoie*/
31
32 int main(int argc, char** argv){
33     int nb;
34     nb=saisieentier();
35     printf("%d\n",nb);
36     return 0;
37 }
```

2 Question 2

```
1 #include <stdio.h> /*Autorise l'emploi de printf et de scanf.*/
2
3 /*! \author Lilian Naretto <narettolil@eisti.eu>
4 \date 15 octobre 2019
5 \file tp3q2.c
6 \brief question 2 du tp3
7 \version 0.1 premier jet*/
8
9 /*! \fn int bidon()
10 \return n, un chiffre
11 \brief affiche un chiffre, demande de saisir un chiffre, le renvoie*/
12
13 int bidon(){
14     int n;
15     printf("%d\n",n);
16     printf("donne un chiffre :");
17     scanf("%d",&n);
18     printf("%d",n);
19 }
20
21 /*! \fn int main(int argc, char** argv)
22 \param argc nombre d'arguments en entrée
23 \param argv valeur des arguments en entrée
24 \return 0 si tout c'est bien passé
25 \brief demande de saisir un chiffre, le renvoie*/
26
27 int main(int argc, char** argv){
28     int n;
29     printf("%d\n",n);
30     printf("donne un chiffre :");
31     scanf("%d",&n);
32     printf("%d",n);
33     bidon();
34     printf("%d\n",n);
35     return 0;
36 }
```

3 Question 3

```
1  #include <stdio.h> /*Autorise l'emploi de printf et de scanf.*/
2
3  /*!\author Lilian Naretto <narettolil@eisti.eu>
4  \date 15 octobre 2019
5  \file tp3q3.c
6  \brief question 3 du tp3
7  \version 0.1 premier jet*/
8
9  /*! \fn int exchange(int nb1,int nb2)
10 \param int nb1
11 \param int nb2
12 \brief échange la valeur de deux variables*/
13
14 int exchange(int nb1,int nb2){
15     int tmp;
16     printf("avant échange\n");
17     printf("nb1=%d\n",nb1);
18     printf("nb2=%d\n",nb2);
19     tmp=nb1;
20     nb1=nb2;
21     nb2=tmp;
22     printf("apres échange\n");
23     printf("nb1=%d\n",nb1);
24     printf("nb2=%d\n",nb2);
25 }
26
27 /*! \fn int main(int argc, char** argv)
28 \param argc nombre d'arguments en entrée
29 \param argv valeur des arguments en entrée
30 \return 0 si tout c'est bien passé
31 \brief échange la valeur de deux variables*/
32
33 int main(int argc, char** argv){
34     int nb1;
35     int nb2;
36     nb1=5;
37     nb2=7;
```

```
33     int main(int argc, char** argv){
34         int nb1;
35         int nb2;
36         nb1=5;
37         nb2=7;
38         printf("avant appel de la fonction\n");
39         printf("nb1=%d\n",nb1);
40         printf("nb2=%d\n",nb2);
41         printf("appel de la fonction échange\n");
42         échange(nb1,nb2);
43         printf("apres appel de la fonction\n");
44         printf("nb1=%d\n",nb1);
45         printf("nb2=%d\n",nb2);
46         return 0;
47     }
```

4 Question 4

```
1  #include <stdio.h> /*Autorise l'emploi de printf et de scanf.*/
2
3  /*!\author Lilian Naretto <narettolil@eisti.eu>
4  \date 15 octobre 2019
5  \file tp3q4.c
6  \brief question 4 du tp3
7  \version 0.1 premier jet*/
8
9  /*!\fn void tablemultiplication(int n)
10 \param int n
11 \brief affiche la table de multiplication de l'entier*/
12
13 void tablemultiplication(int n){
14     int calcul;
15     int i;
16     if (n > 10)
17     { /*traitement du cas où n est superieur a 10*/
18         printf("chiffre trop gros\n");
19     }
20     else
21     { /*sinon, afficher la table*/
22         for (i = 1; i < 10; i++)
23         {
24             calcul = i*n;
25             printf("%d * %d = %d\n",i,n,calcul);
26         }
27     }
28 }
29
30 /*!\fn float fonction(float n)
31 \param float n
32 \brief calcul la fonction voulu pour un n donné*/
33 float fonction(float n){
34     float calcul;
35     calcul= n*n*n + 12*n*n + 1;
36     return calcul;
37 }
```

```
38 /*!\fn double doublesoleequation(void)
39 \param void
40 \brief calcul la solution de l'equation en utilisant la methode dichotomique*/
41 double doublesoleequation(void){
42     float m;
43     float a;
44     float b;
45     a=(-15);
46     b=(-10);
47     while ((b-a) > 0.00001)
48     {
49         m =(a+b)/2;
50         if (fonction(a)*fonction(m)<0)
51         {
52             b=m;
53         }
54         else
55         {
56             a=m;
57         }
58     }
59     return m;
60 }
61
62 /*!\fn int factorielle(int n)
63 \param int n
64 \brief retourne la factorielle d'un chiffre donnée*/
65 int factorielle(int n){
66     int i;
67     int calcul;
68     calcul=1;
69     for (i=1; i <= n; i++)
70     {
71         calcul=calcul*i;
72     }
73     return calcul;
74 }
```

5 Questions 2ème séance

```
1  #include <stdio.h> /*Autorise l'emploi de printf et de scanf.*/
2
3  /*!\author Lilian Naretto <narettolil@eisti.eu>
4  \date 21 octobre 2019
5  \file tp3.c
6  \brief tp3, 2ème séance
7  \version 0.1 premier jet*/
8
9  /*! \fn void affichagesapin(int n)
10 \param int n
11 \brief affiche un arbre selon une longueur donnée*/
12
13 void affichagesapin(int n){
14     int i;
15     int k;
16     if (n<=1) /* si n=1, pas d'arbre */
17     {
18         printf("mauvais chiffre\n");
19     }
20     else /* sinon, on crée l'arbre */
21     {
22         for ( i = 0; i < n; i++)
23         {
24             for ( k = 0; k < 1+2*(n-1); k++)
25             {
26                 if (k >= n-1-i && k <= n-1+i)
27                 {
28                     printf("*");
29                 }
30                 else
31                 {
32                     printf(" ");
33                 }
34             }
35             printf("\n");
36         }
37     }
```

```
39     for ( i = 0; i < 3; i++)
40     {
41         for (k = 0; k < 1+2*n; k++)
42         {
43             if (k >= n-2 && k <= n)
44             {
45                 printf("@");
46             }
47             else
48             {
49                 printf(" ");
50             }
51         }
52         printf("\n");
53     }
54 }
55
56 }
57
58 }
59
60 /*! \fn int factoriel(int n)
61 \param int n
62 \brief retourne la factorielle d'un chiffre donnée*/
63 int factoriel(int n){
64     int i;
65     int k;
66     for (k=1; 1 < n; n--)
67     {
68         k*=n;
69     }
70     return k;
71 }
72 }
```

```
73  /*! \fn int maxfact(int k)
74  \param int k
75  \brief retourne le chiffre qui a sa factorielle majorée par k*/
76  int maxfact(int k){
77      int i;
78      if (k<=1)
79      {
80          printf("mauvais chiffre\n");
81      }
82      for ( i = 1; factoriel(i) < k; i++)
83      {
84      }
85      printf("%d !\n est majorée",i-1);
86
87  }
88
89
90  /*! \fn float puissance(float n, int p)
91  \param float n
92  \param float p
93  \brief donne le chiffre n a la puissance p*/
94
95  float puissance(float n, int p){
96      float a=1;
97      if (p<0)
98      {
99          for (int i = 0; i < -p; i++)
100          {
101              a*=n;
102          }
103          return 1/a;
104      }
105      for (int i = 0; i < p; i++)
106      {
107          a*=n;
108
```

```
109      }
110      return a;
111  }
112
113  }
114
115  /*! \fn void estarmstrong(int n)
116  \param int n
117  \brief indique si n est un chiffre armstrong*/
118
119  void estarmstrong(int n){
120      int a=0;
121      int r= n;
122      for (; r != 0;)
123      {
124          a += puissance(r%10,3);
125          r /= 10;
126      }
127
128      if (n == a)
129      {
130          printf("c'est un chiffre armstrong\n");
131      }
132      else
133      {
134          printf("ce n'est pas un chiffre armstrong\n");
135      }
136
137  }
138
139  }
140
141  /*! void binaire (unsigned int n)
142  \param int n
143  \brief donne la representation binaire d'un chiffre donnée*/
144  void binaire (unsigned int n){
145      int longueur = sizeof(int)*8;
```

```
182  /*! \fn int main(int argc, char** argv)
183  \param argc nombre d'arguments en entrée
184  \param argv valeur des arguments en entrée
185  \brief sert de menu pour choisir les questions voulues*/
186  int main(int argc, char** argv){
187      int n;
188      int p;
189      int question;
190      printf("choisis le numero de ta question : ");
191      scanf("%d",&question);
192      if (question > 5 || question < 0)
193      {
194          printf("mauvais numero\n");
195      }
196      else
197      {
198          switch (question)
199          {
200              case 1:
201                  printf("donne un chiffre : ");
202                  scanf("%d",&n);
203                  affichagesapin(n);
204                  break;
205              case 2:
206                  printf("donne un chiffre : ");
207                  scanf("%d",&n);
208                  maxfact(n);
209                  break;
210              case 3:
211                  printf("donne un chiffre : ");
212                  scanf("%d",&n);
213                  estarmstrong(n);
214                  break;
215              case 4:
216                  printf("donne un chiffre : ");
```

```
219                  break;
220              case 5:
221                  printf("donne deux chiffres n et p : ");
222                  scanf("%d",&n);
223                  scanf("%d",&p);
224                  printf("le coef binomial est : %d\n",coefbin(n,p));
225                  break;
226              default:
227                  break;
228          }
229      }
230
231 }
```