

```
1  //-----
2  // TP 4 Exercice 1
3  #include <stdio.h>
4
5  int main(void) {
6      int n;
7
8      do {
9          printf("entrez un entier : ");
10         scanf("%d",&n);
11         printf("Le caractere de code ASCII %d est %c\n",n,(char) n);
12     } while (n>='a' && n <='z');
13
14     printf("merci et au revoir\n");
15     return 0;
16 }
17
```

```
18
19  //-----
20  // TP 4 Exercice 2
21  #include <stdio.h>
22  #include <math.h>
23
24  int main(void) {
25      double x;
26
27      printf("entrez un reel : ");
28      scanf("%lf",&x);
29
30      while (x>=0) {
31          printf("La racine de %lf vaut %lf\n",x,sqrt(x));
32          printf("entrez un reel : ");
33          scanf("%lf",&x);
34      }
35
36      printf("merci et au revoir\n");
37      return 0;
38 }
39
40 /*autre solution
41 int main(void) {
42     double x;
43
44     while (1) {
45         printf("entrez un reel : ");
46         scanf("%lf",&x);
47         if (x<0)
48             break;
49         printf("La racine de %lf vaut %lf\n",x,sqrt(x));
50     }
51     return 0;
52 }*/
53
```

```
54
55  //-----
56  // TP 4 Exercice 3
57  #include <stdio.h>
58
59  int main(void) {
60      int i,n;
61
62      printf("entrez un nombre entier : ");
63      scanf("%d",&n);
64
65      for (i=n+1;i<=n+10;i++)
66          printf("%d ",i);
67
68      printf("\nmerci et au revoir\n");
69      return 0;
70 }
71
72
73
```

```

74 //-----
75 // TP 4 Exercice 4
76 #include<stdio.h>
77
78 int main(void) {
79     int ligne,colonne,nmax;
80
81     printf("entrez la hauteur du carre : ");
82     scanf("%d",&nmax);
83
84     printf("voici le carre :\n");
85
86     for (ligne=0;ligne<nmax;ligne++) {
87         for (colonne=0;colonne<nmax;colonne++)
88             printf("*");
89         printf("\n"); //on passe a la ligne suivante
90     }
91
92     return 0;
93 }
94
95 //-----
96 // TP 4 Exercice 5
97 #include <stdio.h>
98
99
100 int somme(int); //prototype de la fonction somme
101
102 int main(void) {
103     int n,som; //declaration des variables
104
105     while (1) {
106         // entree des parametres
107         printf("entrez un nombre entier : ");
108         scanf("%d",&n);
109         if (n<=0)
110             break;
111         //appel de la fonction
112         som = somme(n);
113
114         //affichage des resultats
115         printf("la somme des %d premiers entiers vaut %d\n\n",n,som);
116         printf("la vraie valeur est : %d\n", (n*(n+1))/2);
117     }
118
119     printf("merci et au revoir !\n");
120     return 0;
121 }
122
123 int somme(int n) //definition de la fonction somme
124 {
125     int i,s;
126     s=0;
127     for (i=1;i<=n;i++)
128         s+=i;
129     return s;
130 }
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146

```

```

147 //-----
148 // TP 4 Exercice 6
149 #include <stdio.h>
150 #define NMAX 10
151
152 int main(void) {
153     int i,j;
154
155     //on affiche la ligne d'entete
156     printf("      |");
157     for (j=1;j<=NMAX;j++)
158         printf("%4d",j);
159     printf("\n");
160
161     //on affiche la ligne de separation
162     printf("-----|");
163     for (j=1;j<=NMAX;j++)
164         printf("----");
165     printf("\n");
166
167     for(i=1;i<=NMAX;i++) { //boucle sur la ligne
168         printf("%4d |",i); //entete de la ligne
169         for(j=1;j<=NMAX;j++)
170             printf("%4d",i*j); //%4d permet d'afficher le nombre
171                                 //entier sur 4 caracteres exactement
172         printf("\n");
173     }
174     return 0;
175 }
176
177 //-----
178 //TP 4 Exercice 7
179 #include <stdio.h>
180
181 int main(void) {
182     int i,n,x,min,max,somme=0;
183     double moyenne;
184
185     do {
186         printf("combien de valeurs voulez-vous rentrer : ");
187         scanf("%d",&n);
188     } while (n<=0);
189
190     for (i=0;i<n;i++) {
191         printf("valeur n %d : ",i+1);
192         scanf("%d",&x); //on entre une valeur
193         if (i==0) { //au premier tour de boucle
194             min=x; //on initialise le max
195             max=x; //on initialise le min
196         }
197         else if (x<min) min=x; //on remplace le minimum si necessaire
198         else if (x>max) max=x; //on remplace le maximum si necessaire
199         somme+=x; //on incremente la somme des reels
200     }
201     moyenne=(1.*somme)/n; //on calcule la moyenne
202                           //en forçant la division reelle
203
204     //affichage des resultats
205     printf("la valeur moyenne est \t%f\n",moyenne);
206     printf("la valeur minimum est \t%d\n",min);
207     printf("la valeur maximum est \t%d\n",max);
208
209     return 0;
210 }
211
212
213
214
215
216
217
218
219

```

```
220 //-----
221 // TP 4 Exercice 8
222 #include <stdio.h>
223
224 int ispremier (int);
225
226 int main (void) {
227     int n;
228
229     printf("entrez un nombre entier : ");
230     scanf("%d",&n);
231
232     if (ispremier(n))
233         printf("%d est premier\n",n);
234     else
235         printf("%d n'est pas premier\n",n);
236
237     return 0;
238 }
239
240 int ispremier(int n)
241 {
242     int i;
243
244     if (n==1)
245         return 0;    //1 n'est pas premier
246
247     //on recherche un eventuel diviseur de n
248     for (i = 2; i*i<=n; i++) {
249         if (n%i == 0) {
250             return 0;    //on a trouve un diviseur
251         }                //donc le nombre n'est pas premier
252     }
253
254     return 1;    //on n'a trouve aucun diviseur
255                 //donc le nombre est premier
256 }
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
```

```

293 //-----
294 // TP 4 Exercices 9 et 12
295 #include <stdio.h>
296
297 int Suite(int);
298 int Serie_bof(int);
299 int Serie_top(int);
300 void Fibonacci(int);
301
302 int main (void) {
303     int n,u;
304
305     printf("entrez un nombre entier positif : ");
306     scanf("%d",&n);
307
308     u=Suite(n);
309     printf("u_%d = %d\n",n,u);
310     printf("1er calcul : S_%d = %d\n",n,Serie_bof(n));
311     printf("2nd calcul : S_%d = %d\n",n,Serie_top(n));
312
313     printf("\nVoici les %d lers termes de la suite de Fibonacci :\n",n+1);
314     Fibonacci(n);
315     return 0;
316 }
317
318 int Suite(int n)
319 {
320     int i,u;
321     u=1; //on initialise u_0
322     for (i=1;i<=n;i++)
323         u=5*i-2*u; //on calcule u_i
324     return u;
325 }
326
327 int Serie_bof(int n)
328 {
329     int s,i;
330     s=0; //on initialise S à 0
331     for (i=0;i<=n;i++)
332         s=s+Suite(i); //on calcule S_i
333     return s;
334 }
335
336 int Serie_top(int n)
337 {
338     int s,u,i;
339     u=1; //on initialise u_0
340     s=u; //on initialise S_0
341     for (i=1;i<=n;i++){
342         u=5*i-2*u; //on calcule u_i
343         s=s+u; //on calcule S_i
344     }
345     return s;
346 }
347
348 void Fibonacci(int n)
349 {
350     int i,u,u1,u2;
351     u2=1; //u2 représente le terme d'ordre i-2 soit u0 au depart
352     printf("u%2d = %4d\n",0,u2);
353     u1=1; //u1 représente le terme d'ordre i-1 soit u1 au depart
354     printf("u%2d = %4d\n",1,u1);
355
356     for (i=2;i<=n;i++) { //le premier terme a calculer est u2
357         //ob caulcule u_i
358         u=u1+u2;
359         //on prépare le tour de boucle suivant quand i vaudra i+1
360         u2=u1; // u2 devient u_((i+1)-2) soit u1
361         u1=u; // u1 devient u_((i+1)-1) soit u
362         //on affiche u_i
363         printf("u%2d = %4d\n",i,u);
364     }
365 }

```

```

366 //-----
367 //TP 4 Exercice 10
368 #include <stdio.h>
369
370 int main(void) {
371     int x,min,max,somme=0,n=0;
372     double moyenne;
373
374     printf("entrez les nombres positifs un par un\n");
375     printf("jusqu'a entrer un nombre negatif\n");
376
377     scanf("%d",&x); //entree du premier reel
378     min=x; //initialisation du minimum
379     max=x; //initialisation du maximum
380
381     while(x>=0.) {
382         n++; //on incremente le nombre de reels valides
383         if (x<min) min=x; //on remplace le minimum si necessaire
384         if (x>max) max=x; //on remplace le maximum si necessaire
385         somme+=x; //on incremente la somme des reels
386         scanf("%d",&x); //on entre un nouveau reel
387     }
388     moyenne=(1.*somme)/n; //on calcule la moyenne
389
390     //affichage des resultats
391     if (n>0) {
392         printf("la valeur moyenne est \t%lf\n",moyenne);
393         printf("la valeur minimum est \t%d\n",min);
394         printf("la valeur maximum est \t%d\n",max);
395     } else
396         printf("vous n'avez rentre aucun nombre positif\n");
397
398     return 0;
399 }
400
401 //-----
402 // TP 4 Exercice 11
403 #include<stdio.h>
404
405 int main(void) {
406     int ligne,colonne,nmax;
407
408     printf("entrez la hauteur du sapin : ");
409     scanf("%d",&nmax);
410
411     //on affiche le triangle
412     for (ligne=0;ligne<nmax;ligne++) { //boucle sur les lignes
413         for (colonne=0;colonne<(nmax-ligne-1);colonne++)
414             printf(" "); //on affiche les espaces de debut de ligne
415         for(colonne=0;colonne<(2*ligne+1);colonne++)
416             printf("*"); //on affiche les etoiles
417         printf("\n"); //on passe a la ligne suivante
418     }
419
420     //on affiche le tronc
421     for(ligne=0;ligne<nmax;ligne++) { //boucle sur les lignes
422         for(colonne=1;colonne<nmax;colonne++)
423             printf(" "); //espaces de debut de ligne
424         printf("*"); //on affiche une etoile
425         printf("\n"); //on passe a la ligne
426     }
427
428     return 0;
429 }
430
431
432
433
434
435
436
437
438

```

```

439 //-----
440 //TP 4 Exercice 13
441 #include <stdio.h>
442 #include <stdlib.h>
443 #include <time.h>
444 #include <math.h>
445
446 #define R RAND_MAX //coté du carré et rayon du cercle
447
448 int CoordAleat(void);
449 int InCercle(int x,int y,int r);
450 double PiProba(int);
451
452 int main(void) {
453     int n;
454     double pi;
455
456     srand(time(0));
457
458     printf("nombre de points : ");
459     scanf("%d",&n);
460
461     pi = PiProba(n);
462     printf("valeur de pi estimee : \t%f\n",pi);
463     printf("valeur de pi exacte : \t%f\n", M_PI);
464     printf("difference : %g\n",pi-M_PI);
465
466     return 0;
467 }
468
469 //fonction tirant un nombre au hasard entre -R/2 et +R/2
470 int CoordAleat(void)
471 {
472     return rand()-R/2.;
473 }
474
475 //fonction indiquant si un point de coordonnees {x,y} est situé
476 //à l'intérieur du cercle de rayon r centré en {0,0}
477 int InCercle(int x,int y,int r)
478 {
479     if ((x*x+y*y)<=(r*r))
480         return 1;
481     else
482         return 0;
483 }
484
485 //fonction déterminant une valeur approchée de pi de façon probabiliste
486 //on tire 'n' points au hasard dans un carré et on compte le nombre
487 //de points situés à l'intérieur du cercle inscrit : le rapport de
488 //ces 2 nombres tend vers pi/4
489 double PiProba(int n)
490 {
491     int x,y,i,nc;
492
493     nc=0;
494     for (i=0;i<n;i++) {
495         x=CoordAleat();
496         y=CoordAleat();
497         if (InCercle(x,y,R/2.))
498             nc++;
499     }
500     return (4.*nc/n);
501 }
502

```