

```

1  //-----
2  //TP 3 Exercice 1
3
4  #include <stdio.h>
5  int div7(int);
6
7  int main(void) {
8      int n;
9
10
11     printf("Entrez un nombre entier :\n");
12     scanf("%d",&n);
13
14     if(div7(n))
15         printf("%d est divisible par 7\n",n);
16     else
17         printf("%d n'est pas divisible par 7\n",n);
18
19     return 0;
20 }
21
22 //fonction renvoyant 1 si l'entier passé en argument est divisible par 7
23 //et renvoyant 0 sinon
24 int div7(int n)
25 {
26     return !(n%7);
27 }
28
29
30 //-----
31 //TP 3 Exercice 2
32
33 #include <stdio.h>
34
35 int main(void) {
36     int a,b;
37
38     //le plus grand ?
39     printf("Entrez 2 nombres :\n");
40     scanf("%d",&a);
41     scanf("%d",&b);
42
43     if(a==b)
44         printf("a et b sont egaux\n"); //cas a==b
45     else if (a<b)
46         printf("%d est plus grand que %d\n",b,a); //cas b>a
47     else
48         printf("%d est plus grand que %d\n",a,b); //cas a>b
49
50     return 0;
51 }
52
53
54
55 //-----
56 // TP 3 Exercice 3
57 #include <stdio.h>
58
59 int main(void) {
60     char c;
61
62     printf("entrez une lettre : ");
63     scanf("%c",&c);
64
65     // on verifie que c est une lettre
66     if ((c>='a' && c<='z') || (c>='A' && c<='Z'))
67         printf("C'est une lettre !\n");
68     //on verifie si c'est un chiffre
69     else if (c>='0' && c<='9')
70         printf("C'est un chiffre !\n");
71     else
72         printf("Ce n'est ni une lettre ni un chiffre !\n");
73

```

```

74     //code ASCII pair ?
75     printf("le code ASCII (%d) de %c est ",c,c);
76     if (c%2==0)
77         printf("pair\n");
78     else
79         printf("impair\n");
80
81     return 0;
82 }
83
84
85 //-----
86 //TP 3 Exercice 4
87 #include <stdio.h>
88
89 //programme principal
90 int main(void) {
91     int NHT, NHS;
92     double TH,ST;
93
94
95     printf("Entrez le nombre d'heures travaillees : ");
96     scanf("%d",&NHT);
97     printf("Entrez le taux horaire : ");
98     scanf("%lf",&TH);
99
100     if (NHT>37) {
101         NHS=NHT-37;
102         ST=TH*(NHT+0.25*NHS);
103     } else {
104         NHS=0;
105         ST=TH*NHT;
106     }
107
108     printf("vous avez fait %d heures supplementaires\n",NHS);
109     printf("votre salaire net hebdomadaire est de %.2f euros\n",ST);
110
111     return 0;
112 }
113
114
115 //-----
116 // TP 3 Exercice 5
117 #include <stdio.h>
118
119 int max3(int,int,int);
120
121 int main(void) {
122     int a,b,c;
123
124     //entree des variables
125     printf("Entrez 3 nombres :\n");
126     scanf("%d",&a);
127     scanf("%d",&b);
128     scanf("%d",&c);
129
130     printf("le plus grand des 3 nombres est %d\n",max3(a,b,c));
131
132     return 0;
133 }
134
135 //fonction max3 renvoyant la plus grandes des 3 valeurs passees comme argument
136 int max3(int a,int b,int c)
137 {
138     if(a>b && a>c) //a est le plus grand
139         return a;
140     else if (b>c) //sinon b est le plus grand
141         return b;
142     else //sinon c est forcement le plus grand
143         return c;
144 }
145
146

```

```

147 //-----
148 // TP 3 Exercices 6 et 11
149 #include <stdio.h>
150 #include <math.h>
151
152 int main(void) {
153     //declaration des variables
154     double a,b,c;           //coefficients de l'equation
155     double delta;           //discriminant
156     double x0,x1,x2;         //racines reelles
157     double reel,imag;        //racines complexes
158
159     //entree des parametres
160     printf("Resolution de l'equation a.x^2+b.x+c=0\n");
161     printf("Entrez le coefficient a :");
162     scanf("%lf",&a);
163     printf("Entrez le coefficient b :");
164     scanf("%lf",&b);
165     printf("Entrez le coefficient c :");
166     scanf("%lf",&c);
167
168     //resolution de l'equation
169     if (a!=0) { //on teste si l'equation est vraiment du second degre
170         delta = b*b-4*a*c; //calcul du discriminant
171
172         if (delta > 0) { // cas ou il y a deux racines reelles
173             x1=(-b-sqrt(delta))/(2.*a); //premiere racine
174             x2=(-b+sqrt(delta))/(2.*a); //seconde racine
175             printf("il y a deux racines reelles : x1 = %f et x2 = %f\n",x1,x2);
176         }
177         else if (delta == 0) { //cas ou il y a une seule racine
178             x0=-b/(2.*a);
179             printf("il y a une racine reelle : x0 = %f\n",x0);
180         }
181         else { //cas ou il y a deux racines complexes
182             reel=-b/(2.*a); //partie reelle
183             imag=fabs(sqrt(-delta)/(2.*a)); //partie imaginaire
184             printf("il y a deux racines complexes : \n");
185             printf("x1 = %f + %f i\nx2 = %f - %f i\n",reel,imag,reel,imag);
186         }
187     }
188     //on arrive aux cas ou a=0
189     else if (b != 0) { //on a une equation du premier degre
190         x0=-c/b;
191         printf("il y a une racine reelle : x0 = %f\n",x0);
192     }
193
194     else if (c!=0) //seul c est non nul
195         printf("il n'y a pas de solution\n");
196
197     else //l'equation est 0 = 0
198         printf("toute valeur complexe est solution\n");
199
200     return 0;
201 }
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219

```

```

220 //-----
221 // TP 3 Exercice 7
222 #include <stdio.h>
223
224 int isbissextile (int);
225 int isbissextile2 (int);
226 int isbissextile3 (int);
227
228 int main(void) {
229     int annee;
230
231     //entree des variables
232     printf("Entrez l'annee :");
233     scanf("%d",&annee);
234
235     //version avec tests imbriqués et drapeau
236     if (isbissextile(annee))
237         printf("%d est bissextile\n",annee);
238     else
239         printf("%d n'est pas bissextile\n",annee);
240
241     //version sans aucun test
242     if (isbissextile3(annee))
243         printf("%d est bissextile\n",annee);
244     else
245         printf("%d n'est pas bissextile\n",annee);
246
247     return 0;
248 }
249
250 //fonction determinant si l'année passée en argument est bissextile
251 //version avec un drapeau
252 int isbissextile (int annee)
253 {
254     int flag = 1; //on présume que l'année est bissextile
255     if (annee%4!=0) //l'année n'est pas divisible par 4
256         flag=0;
257     else
258         if (annee%100==0) //l'année est divisible par 100
259             if (annee%400!=0) //mais pas par 400
260                 flag=0;
261
262     return flag;
263 }
264
265 //fonction determinant si l'année passée en argument est bissextile
266 //version avec un seul test
267 int isbissextile2 (int an)
268 {
269     if(an%4 == 0 && (an%100 != 0 || an%400 == 0))
270         return 1;
271     else
272         return 0;
273 }
274
275 //fonction determinant si l'année passée en argument est bissextile
276 //version sans aucun if
277 int isbissextile3 (int an)
278 {
279     return (an%4 == 0 && (an%100 != 0 || an%400 == 0));
280 }
281
282
283
284
285
286
287
288
289
290
291
292

```

```

293 //-----
294 // TP3 Exercice 8
295 #include <stdio.h>
296
297 //prototype de la fonction
298 double Impots_Bordures(double RT, int N, int A, int R);
299
300
301 int main(void) {
302     int N; //nombre de personnes dans le foyer
303     double RT; //Revenu total
304     int PA; //présence de personne agee (1:oui / 0:non)
305     int FR; //Personne de la famille royale (1:oui / 0:non)
306     double IR; //impots sur le revenu
307
308     //entree des variables
309     printf("Entrez le nombre de personnes : ");
310     scanf("%d",&N);
311     printf("entrez le revenu total : ");
312     scanf("%lf",&RT);
313     printf("personnes agees dans le foyer (1:oui/0:non) : ");
314     scanf("%d",&PA);
315     printf("membres de la famille royale dans le foyer (1:oui/0:non) : ");
316     scanf("%d",&FR);
317
318     //appel de la fonction
319     IR=Impots_Bordures(RT,N,PA,FR);
320
321     //affichage du resultat
322     printf("l'impot a payer est de %.2f couronnes bordures\n",IR);
323
324     return 0;
325 }
326
327 //définition de la fonction
328 double Impots_Bordures(double RT, int N, int A, int R)
329 {
330     double IR, RPP, NP;
331
332     //determination du nombre de parts NP
333     if (N<=2)
334         NP=N;
335     else
336         NP=2+(N-2)/2.;
337
338     //calcul du Revenu par part RPP
339     RPP=RT/NP;
340
341     //calcul de l'impot avant abattement IR
342     if (RPP<2000)
343         IR = 0; //exonération d'impots jusqu'à 2000
344     else if (RPP<3000)
345         IR = 0.1*(RPP-2000); //on impose a 10% ce qui dépasse 2000
346     else if (RPP<5000)
347         IR = 100+0.2*(RPP-3000); //on impose à 20% ce qui dépasse 3000
348         //et à 10% ce qui est entre 2000 et 3000 soit 100
349     else
350         IR = 500+0.5*(RPP-5000); //on impose à 50% ce qui dépasse 5000
351         //à 20% ce qui est entre 3000 et 5000 soit 400
352         //et à 10% ce qui est entre 2000 et 3000 soit 100
353
354     //on remultiplie par le nombre de parts
355     IR=IR*NP;
356
357     //on applique les abattements éventuels
358     if (R) //en 1er l'abattement le plus intéressant
359         IR=0.85*IR;
360     else if (A) //puis le 2nd abattement uniquement si
361         IR=0.95*IR; //on n'a pas déjà appliqué le 1er
362
363     return IR;
364 }
365

```

```

366
367 //-----
368 // TP 3 Exercices 9 et 10
369 #include <stdio.h>
370
371 int isbissextile (int); //fonction indiquant si l'année est bissextile
372 int isDateValide(int a,int m,int j); //fonction indiquant si la date est valide
373 int nbjour(int a,int m); //fonction renvoyant le nombre de jours dans le
    mois
374
375 int main(void) {
376     int j,m,a; //jour, mois,annee
377
378     //entree de la date
379     printf("Entrez la date sous la forme jour mois annee :");
380     scanf("%d %d %d",&j,&m,&a);
381
382
383     //on s'arrete si la date n'est pas valide
384     if (!isDateValide(a,m,j)) {
385         printf("la date entree n'existe pas\n");
386         return 0; //permet de quitter le programme
387     }
388
389
390     //on determine la date du lendemain
391     if (j==nbjour(a,m)) { //en fin de mois, on change de mois et on repasse le jour a 1
392         j=1;
393         if (m==12) { //on change d'annee et on repasse en janvier
394             m=1;
395             if (a==--1)
396                 a=1; //l'année 0 n'existe pas, on passe à l'année 1
397             else
398                 a=a+1; //on passe à l'année suivante
399         }
400         else //on change de mois dans l'annee
401             m=m+1;
402     }
403     else //on change de jour dans le mois
404         j=j+1;
405
406     //affichage de la date du lendemain
407     printf("demain, nous serons le %02d/%02d/%d\n",j,m,a); // le %02d permet
408         //d'afficher les entiers sur exactement 2 caracteres
409
410     return 0;
411 }
412
413 //fonction renvoyant le nombre de jour dans le mois m de l'annee a
414 int nbjour(int a, int m)
415 {
416     if (m==1 || m==3 || m==5 || m==7 || m==8 || m==10 || m==12)
417         return 31; //mois de 31 jours
418     else if (m==4 || m==6 || m==9 || m==11)
419         return 30; //mois de 30 jours
420     else if (isbissextile(a))
421         return 29; //mois de fevrier et annee bissextile
422     else
423         return 28; //mois de fevrier et annee non bissextile
424 }
425
426 //fonction vérifiant que la date est valide
427 int isDateValide(int a,int m,int j)
428 {
429     if (a==0) //l'annee 0 n'existe pas
430         return 0;
431     else if (m<1 || m>12) //le mois n'existe pas
432         return 0;
433     else if (j<1 || j>nbjour(a,m)) //le jour n'existe pas
434         return 0;
435     else
436         return 1; // la date est valide
437 }

```

```
438
439 //fonction determinant si l'année passée en argument est bisextile
440 int isbissextile (int an)
441 {
442     if(an%4 == 0 && (an%100 != 0 || an%400 == 0))
443         return 1;
444     else
445         return 0;
446 }
447
```