

 DEVUIP Service Scolarité	ANNEE UNIVERSITAIRE 2011-2012 SESSION 2 D'AUTOMNE Parcours : LPI UE : M1PY3002EXI Epreuve : Informatique pour la physique Date : Jeudi 07 Juin 2012 Heure : 11h00 Durée : 1h30 Documents : aucun document autorisé Epreuve de M. Dumora	
--	---	---

Chacune des sections est indépendante de la précédente.

1 Exercice I (6 points)

Cet exercice propose d'analyser le fichier source suivant puis de le modifier afin d'introduire de nouvelles fonctionnalités au programme.

1-1 Analyse

1-1-a

L'exécutable affiche à l'écran :

```
256 48
208 48
160 48
112 48
64 48
16 48
16 32
16 16
```

256/48 = 16/3

1-1-b

Il réduit la fraction $\frac{a}{b}$

1-2 Modification du code

1-2-a

```
1  /* Exam M1PY3002EXI
2     seconde session
3     7 Juin 2012
4     programme2
5     auteur: Denis Dumora
6  */
7
8  #include <stdio.h>
9
10 int main(void)
```

```

11 {
12     int a = 256;
13     int b = 48;
14     int p;
15     int n;
16     printf("Entrez le numérateur puis le dénominateur de la fraction à réduire\n");
17     scanf("%d%d",&a,&b);
18     p = a;
19     n = b;
20     printf("%d%d\n",p,n);
21     while(p!=n)
22     {
23         if (p>n)
24         {
25             p = p - n;
26         }
27         else
28         {
29             n = n - p;
30         }
31         printf("%d%d\n",p,n);
32     }
33     printf("-----\n");
34     printf("%d/%d=%d/%d\n",a,b,a/n,b/n);
35     return 0;
36 }

```

1-2-b

```

1  /* Exam M1PY3002EXI
2     seconde session
3     7 Juin 2012
4     programme3
5     auteur: Denis Dumora
6  */
7
8  #include <stdio.h>
9
10 int main(void)
11 {
12     int a = 256;
13     int b = 48;
14     int p;
15     int n;
16     FILE *fp;
17     printf("Entrez le numérateur puis le dénominateur de la fraction à réduire\n");
18     scanf("%d%d",&a,&b);
19     p = a;
20     n = b;
21     fp = fopen("resultat.dat","w");

```

```

22  if (fp==NULL)
23      {
24          printf("Impossible_d'ouvrir_le_fichier\n");
25          return -1;
26      }
27  printf("%d_%d\n",p,n);
28  while(p!=n)
29      {
30          if (p>n)
31              {
32                  p = p - n;
33              }
34          else
35              {
36                  n = n - p;
37              }
38          printf("%d_%d\n",p,n);
39      }
40  printf("-----\n");
41  printf("%d/%d_=%d/%d\n",a,b,a/n,b/n);
42  fprintf(fp,"%d/%d_=%d/%d\n",a,b,a/n,b/n);
43  fclose(fp);
44  return 0;
45  }

```

2 Chute d'une bille dans la glycérine (14 points)

2-1 Eléments de code et compilation

2-1-a

Ces deux lignes sont des directives de précompilation, elles signifient qu'avant la compilation du code source doivent être inclus dans le code les deux fichiers `stdio.h` et `math.h`, ces fichiers contiennent respectivement les déclarations des fonctions d'entrée/sortie standard et les fonctions mathématiques.

2-1-b

```
gcc chute_glycerine.c -o chute_glycerine -Wall -lm
```

2-2 Modification du code

2-2-a

```

float nouvelle_vitesse(float vi, float force, float masse, float dt)
{
    float v;
    v = force/masse*dt+ vi;
    return v;
}

float vexact(float t, float g, float masse, float coefstokes)
{

```

```

float v;
v = -g*masse/coefstokes*(1-exp(coefstokes/masse*t));
return v;
}

```

2-2-b

```

/* Boucle sur le calcul de la vitesse */

for(t=tini;t<=tfin;t=t+dt)
{
    float v_exact = vexact(t, g, m, coef_stokes);

    float f_stokes = coef_stokes*v;
    float f = m*g + f_stokes;
    v = nouvelle_vitesse(v,f, m, dt);
}

/* Fin du calcul de la vitesse par la méthode d'Euler */

```

2-2-c

```

/* Exam M1PY3002EXI
   seconde session
   7 Juin 2012

chute d'une bille dans de la glycerine

@author: dumora */

#include <stdio.h>
#include <math.h>

float nouvelle_vitesse(float vi,float force, float masse, float dt)
{
    float v;
    v = force/masse*dt+ vi;
    return v;
}

float vexact(float t, float g, float masse, float coefstokes)
{
    float v;
    v = -g*masse/coefstokes*(1-exp(coefstokes/masse*t));
    return v;
}

int main(void)
{

```

```

/* donnees physique */
/* rayon de la bille en metre */
float r = 4e-4;
/* viscosite de la glycerine */
float eta = 1.5;
/* masse de la bille en kg*/
float m = 2e-3;
float g = 9.81;
float pi = acos(-1.);

/* Initialisations */
float coef_stokes = -6.*pi*eta*r;
float vini = 0.;
float tini = 0.;
float tfin = 1.;
/* nombre de points sur l'axe des temps */
float Npt = 20;
/* calcul de dt */
float dt = (tfin-tini)/(Npt-1);
int i;
float v;
float t;

v = vini;

printf("temps_v_Euler_v_exacte_ecart_relatif\n");
/* Boucle sur le calcul de la vitesse */
for(t=tini;i<=tfin;t=t+dt)
{
    float v_exact = vexact(t, g, m, coef_stokes);
    float ecart_pourcent = (v-v_exact)/v_exact*100.;
    printf("%.3f_%.3f_%.3f_%.2f%%\n",t,v,v_exact,ecart_pourcent);

    float f_stokes = coef_stokes*v;
    float f = m*g + f_stokes;
    v = nouvelle_vitesse(v,f, m, dt);
}
/* Fin du calcul de la vitesse par la methode d'Euler */
return 0;
}

```

2-3 Analyse du résultat

2-3-a

Au départ, les écarts entre la vitesse au début et à la fin de l'intervalle de temps considéré est plus important qu'à la fin du mouvement étudié pour lequel la vitesse de la bille est proche de la vitesse limite de chute de la bille.

L'erreur commise lorsqu'on confond dérivée et accroissement de vitesse au début de la chute est donc plus grande que lorsqu'on s'approche de la vitesse limite. D'où un écart relatif plus grand entre vitesse calculée par la méthode d'Euler et vitesse exact en début de chute.

FIN...