

Epreuve d'informatique, octobre 2017

Pour répondre à une question, cocher la case correspondante dans le tableau de la page 1.

- 1 seule réponse possible par question,
- Une bonne réponse = +1
- Une mauvaise réponse = -1 (avec malus maximal: -2).

Réponse	a	b	c	d	e
1.1 Bash 1					
1.2 Bash 2					
1.3 Bash 3					
1.4 Bash 4					
2 Syntaxe					
3 Variables et fonctions					
4 Fonctions					
5 Passage de paramètres					
6 Fichiers 1					
7 Fichiers 2					
8 Boucles 1					
9 Boucles 2					
10 Boucles et tableaux					
11 Algorithme					

NOTA BENE: Pour tous les programmes, on fait abstraction des fichiers d'en-tête. Vous n'avez donc pas à vérifier si les `#include` sont correctement écrits...

1 Commandes bash

Dans notre répertoire personnel `/MAISON`, on a deux répertoires `/MAISON/CAVE` et `/MAISON/GRENIER`. On a aussi les répertoires `/MAISON/CAVE/CONGEL` et `/MAISON/GRENIER/PLACARD`. Le `PLACARD` est plein car il y a un fichier `/MAISON/GRENIER/PLACARD/LIVRE.txt`. On a aussi un fichier qui traîne, mal rangé : `/MAISON/GLACE.txt`

1.1 Bash 1

Quelles sont les commandes qui permettent de ranger le fichier `GLACE.txt` à sa place, sachant que l'on se trouve dans le répertoire `/MAISON` ?

a `mv GLACE.txt /CAVE/CONGEL`

b `cd CAVE`
`cd CONGEL`
`mv GLACE.txt`

c `mv GLACE.txt CAVE/CONGEL`

d `cd CAVE`
`mv ../GLACE.txt /CONGEL`

1.2 Bash 2

On se trouve dans `/MAISON/CAVE`. Quelles sont les commandes qui permettent de vider le `PLACARD` ?

a `cd ..`
`rm /GRENIER/PLACARD/LIVRE.txt`

b `cd ../../GRENIER/PLACARD`
`rm /LIVRE.txt`

c `rm LIVRE.txt`

d `cd ..`
`cd GRENIER/PLACARD`
`rm LIVRE.txt`

1.3 Bash 3

On voudrait créer un répertoire `/MAISON/SALON` et avec dedans, un répertoire `/MAISON/SALON/ARMOIRE`. Quelles sont les commandes qui permettent de faire cela, sachant que l'on se trouve dans `/MAISON/CAVE`.

a) `mkdir SALON`
`cd SALON`
`mkdir ARMOIRE`

b) `cd ..`
`mkdir SALON`
`mkdir SALON/ARMOIRE`

```
c) cd ..  
    mkdir SALON  
    mkdir ARMOIRE
```

1.4 Bash 4

On se trouve dans /MAISON/GRENIER. On tape `cd /PLACARD`.
L'ordinateur :

```
1) est content  
2) renvoie "no such file or directory"
```

Si on tape `cd PLACARD`. L'ordinateur :

```
3) est content  
4) renvoie "no such file or directory"
```

a : (1-3), b :(1-4), ,c :(2-4) d :(2-3)

2 Syntaxe

On veut écrire un programme permettant de comparer deux méthodes numériques de calcul d'intégrale. Cependant, un certain nombre d'erreurs se sont glissées. Dans quelles lignes se trouvent-elles?

```
1 #include <iostream>  
   using namespace std;  
2 #include <math>  
3 int N  
4 double rectangle(double xmax)  
5 {  
6   double dx, integ=0;  
7   dx= xmax/N;  
8   for (int i=1, i<=N, i++)  
9     integ= integ + cos(dx*i);  
10  return integ;  
11 }  
12 void trapeze(double xmax)  
13 {  
14   double dx=xmax/N, integ=0;  
15   do (int i=1; i<=N-1; i++)  
16     integ= integ + 0.5*(cos(dx*i) + cos(dx*(i+1)));  
17   return integ;  
18 }
```

```

19 void main()
20 {double theta;
21  cout<<"Donnez la borne superieure de l'integrale "<<;
22  cin>>theta;
23  cout<<"Indiquez le nombre d'intervalles;
24  cin>>N
25  double x1= rectangle(theta);
26  double x2= trapeze(theta);
27  cout<<"Premiere methode:  "<<x1<<" Deuxieme methode:"<<x2
28  }

```

- (a) 3,4,12,14,16,23,24,25,26
- (b) 8,9,15,16,21,23,27
- (c) 1,2,3,9,10,14,16,17,21,24,27
- (d) 12,15,16,21,22,25,26
- (e) 3,8,12,15,21,23,24,27

3 Variables et fonctions

Qu'affiche le programme suivant :

```

#include <iostream>
using namespace std;
int p=4 ;
//+++++++
int trans(int & r)
{
    int cloc=0, pomp=5 ;
    r=r+1 ;
    cloc=cloc+pomp+r ;
    return (cloc) ;
}
//+++++++
void main ( )
{
    int bla=6, pomp=3 ;
    cout<<pomp <<" ";
    cout<<bla<<" ";
    cout<<trans(bla)<<" ";
    cout<<bla<<" ";
    cout<<pomp<<endl ;
}

```

a) 3 6 12 6 3 b) 3 6 11 6 3 c) 3 6 11 7 3 d) 3 6 11 7 5 e) 3 6 12 7 3

4 Fonctions

Dans le programme ci-dessous, on veut rajouter deux fonctions: "init()" qui initialise les valeurs du tableau T et de x, puis "F()" qui calcule y a partir de T et de x. Comment doivent-elles être déclarées?

```
#include <iostream>
using namespace std;
void main()
{
    double x,y; double T[4]; int error= 0;
    init(T,x);
    while (error==0)
    {
        error= F(T,x,y);
        x= x/2.;
    }
    cout<<"Resultats en x= "<<x<<" y(x)= "<<y<<endl;
}
```

- (a) void init(double &T[4], double &x)
 int F(double &T[4], double &x, double &y)
- (b) void init(double T[4], double &x)
 void F(double T[4], double *x, double &y)
- (c) void init(double *T, double *x)
 int F(double *T, double x, double *y)
- (d) void init(double T[4], double &x)
 int F(double T[4], double x, double &y)
- (e) void init(double T[4], double x)
 int F(double T[4], double x, double &y)

5 Passage de paramètres

Dans la fonction ci-dessous,

```
? evolution( ? )
{
    if (test(t)==0)
```

```

    {
        propage(t,y);
        affiche(y);
        return(0);
    }
    return(1);
}

```

seules certaines valeurs de la variable temps `t` vont donner lieu a un calcul de propagation du type Runge-Kutta. Ce calcul est effectué dans la fonction "`propage()`", qui renvoie la nouvelle valeur de `t` ainsi que les 3 grandeurs calculées, stockées dans le tableau `y`. Les deux premières grandeurs sont ensuite affichées a l'écran. Comment doivent être déclarées les fonctions "`evolution()`", "`test()`", "`propage()`" et "`affiche()`" ?

- (a) `int evolution(double &t, double &y[3])`
`void test(double t)`
`void propage(double &t, double &y[3])`
`void affiche(double y[2])`
- (b) `int evolution(double t, double y[2])`
`int test(double t)`
`void propage(double t, double y[2])`
`void affiche(double y[2])`
- (c) `int evolution(double &t, double *y)`
`void test(double t)`
`void propage(double t, double *y)`
`void affiche(double *y)`
- (d) `int evolution(double &t, double y[3])`
`int test(double t)`
`void propage(double &t, double y[3])`
`void affiche(double y[3])`
- (e) `void evolution(double *t, double y[3])`
`int test(double t)`
`void propage(double *t, double y[3])`
`void affiche(double y[2])`

6 Fichiers 1

Voici un programme:

```

#include <fstream>
#include <iostream>
using namespace std;
void main( )
{
    ifstream f("donnees.dat");
    ofstream g0("sort0.dat");
    ofstream g1("sort1.dat");
    ofstream g2("sort2.dat");
    int v;
    while (f>>v)
    {
        if (v%2==0 && v%3==0)
            g0<<v<<endl;
        if (v%3==0)
            g1<<v<<endl;
        if (v%3==0 || v%5==0)
            g2<<v<<endl;
    }
    f.close( );
    g0.close( );
    g1.close( );
    g2.close( );
}

```

Le fichier "donnees.dat" contient les valeurs suivantes: 48, 6, 21, 15, 5, 36, 16, 260, 10, 216, 125. Que contiennent les fichiers de sortie?

- (a) sort0.dat: 48,6,21,15,36,16,260,10,216
 sort1.dat: 6,21,15,36
 sort2.dat: 48,6,21,15,5,36,260,10,216,125
- (b) sort0.dat: 6,36
 sort1.dat: 6,21,15,36,216
 sort2.dat: 6,21,15,5,36,260,10,125
- (c) sort0.dat: 48,6,36,216
 sort1.dat: 48,6,21,15,36,216
 sort2.dat: 48,6,21,15,5,36,260,10,216,125
- (d) sort0.dat: 48,6,36,216
 sort1.dat: 6,21,15,36,216
 sort2.dat: 6,21,15,5,36,260,10,125

- (e) sort0.dat: 48,6,36,16,216
sort1.dat: 48,6,21,15,36,216
sort2.dat: 6,21,15,5,36,260,10,125

7 Fichiers 2

Qu'affiche le programme suivant:

```
#include <iostream>
using namespace std;
#include <fstream>
void main()
{ int i,boum,blop ;
  ofstream fic1("albert.dat") ;
  fic1<<9 <<endl;
  for (i=1 ;i<10 ; i++)
    fic1<<i<<" "<<i*i<<endl ;
  fic1.close() ;
  ifstream fic2("albert.dat") ;
  fic2>>boum ;
  boum=boum/2;
  for (i=1 ;i<=boum ;i++)
    { fic2>>blop ; cout<<blop<<" "<<flush; }
}
```

- a) 1 2 3 4 b) 1 1 2 4 c) 1 1 2 4 3 9 4 16 d) 1 2 3 4 5 e) 1 4 9 16

8 Boucle 1

Qu'affiche le programme suivant?

```
#include <iostream>
using namespace std;
main(
{
  int r=1;
  for (int i=1;r!=0;i=i+1)
  {
    r=i%3; // signifie r = i modulo 3
    cout<<i<<" "<<r<<" "<<flush;
  }
}
```


- a) 1 1 2 2 3 0 b) 1 0 2 0 3 0 c) la boucle ne s'arrête pas d) 1 1 2 2
 e) 1 3 2 6 3 9

9 Boucle 2

Qu'affiche le bout de programme suivant:

```
i=3;
while (i<3)
{
    cout<<i<<" ";
    i=i+1;
}
```

- a/ il n'affiche rien b/ 1 2 3 c/ 3 d/ 1 2 e/ 1

10 Boucles et tableau

On souhaite créer un tableau, puis l'afficher à l'écran de manière à afficher à l'écran :

```
11 12 13 14 15
21 22 23 24 25
31 32 33 34 35
```

Lequel des programmes suivant est le bon (compile et affiche le résultat désiré) ?

a)

```
int main()
{
    int tableau[5*3];
    int *p=tableau;
    for(int i=0;i<3; i++) for(int j=0;j<5;j++) {*p=(i+1)*10+j;p++}
    cout <<tableau<<endl;
}
```

b)

```
int main()
{
    int tableau[5*3];
    int *p=tableau;
    for(int i=0;i<3; i++)
        for(int j=1;j<=5;j++)
            {*p=(i+1)*10+j;p++;}

    for(int i=0;i<3; i++)
    {
```

```

        for(int j=0;j<5;j++) {cout << tableau[i*5+j]<<" ";}
        cout <<endl;
    }
}
c)
int main()
{
    int tableau[5*3];
    for(int i=0;i<3; i++)
        for(int j=1;j<5) {tableau[i*5+j]=(i+1)*10+j;}

    for(int i=0;i<15;i++) cout << tableau[i];
}
d)
int main()
{
    int tableau[5*3];
    for(int i=0;i<3; i++)
        for(int j=1;j<=5){tableau[i*5+j]=(i+1)*10+j;}

    for(int i=0;i<15;i++) cout << tableau[i];
}

```

11 Algorithme

Qu'affiche le programme:

```

#include <math>
#include <iostream>
using namespace std;
double f(double x)
{
    return cos(x/2.0);
}
//-----
main()
{
    double a=0,b=4;
    while((b-a)>0.001)
    {
        double c=(a+b)/2;
        if(f(c)>0)

```

```
        a=c;  
    else  
        b=c;  
    }  
    cout<<b<<endl;  
}
```

a) 0.001 b) 3.1416 c) 1.5708 d) 1.0000 e) 6.2832