



Langage C++

Interrogation n° 2

R. Absil (abs)
6 décembre 2017

/175

Consignes générales

1. Notez votre nom, prénom, groupe et matricule sur chacune des feuilles que vous remettez à votre maître-assistant, y compris les brouillons.
2. Rédigez vos réponses en français correct : soignez votre orthographe et votre grammaire.
3. Justifiez chacune de vos réponses de façon complète et concise : une réponse incorrecte justifiée avec de bons arguments peut rapporter quelques points, alors qu'une bonne réponse non justifiée sera systématiquement comptée comme nulle.
4. L'interrogation est à cahier fermé.
5. L'interrogation dure 2h.

Question 1. On souhaite que les codes suivants produisent l'affichage souhaité : corrigez chacun d'entre eux pour que ce soit le cas. Justifiez *également* les raisons pour lesquelles l'affichage souhaité apparaît ou non.

/130

Remarques :

- ces codes ne comprennent que des erreurs sémantiques et / ou de logique (ne cherchez donc pas un « ; » ou un « `std::` » manquant);
- il est possible qu'il soit impossible de produire l'affichage souhaité, si tel est le cas, justifiez pourquoi;
- vous ne *pouvez pas* changer de déclaration ou d'affectation de variable, ou de définition de prototype, à moins que cela soit la seule solution possible pour produire l'affichage souhaité;
- vous ne pouvez pas supprimer des lignes de code, à moins que cela soit la seule solution possible pour produire l'affichage souhaité;
- sauf mention contraire, ces codes sont exécutés sur un processeur 64 bits de la famille x86, où 1 byte = 8 bits;
- chaque code vaut 10 points, il y a 13¹ codes;
- répondez sur le papier ministre, vous pouvez référencer le code via les numéros de ligne.

1. Et 13, ça porte malheur ☹. Joyeuse S^t Nicolas !

Matricule :

Nom :

Prénom :

Groupe :

Code 1

```
1 #include <iostream>
2
3 class Integer
4 {
5     unsigned i;
6     bool positive;
7
8     public:
9         Integer(unsigned i, bool positive = true) : i(i), positive(positive) {}
10        Integer operator +(Integer a)
11        {
12            if(positive && a.positive)
13                return Integer(i + a.i);
14            else if(positive && !a.positive)
15                return Integer(i - a.i);
16            else if(!positive && a.positive)
17                return Integer(a.i - i);
18            else
19                return Integer(-i - a.i);
20        }
21
22        friend std::ostream& operator << (std::ostream& out, const Integer& i)
23        {
24            if(!i.positive)
25                out << "-";
26            out << i.i;
27        }
28    };
29
30    int main()
31    {
32        Integer a1(2u);
33        Integer a2(3u, false);
34
35        Integer s = a1 + a2;
36        std::cout << s << std::endl;
37    }
```

Affichage souhaité du Code 1

1 -1

Affichage produit par le Code 1

1 4294967295

Matricule :

Nom :

Prénom :

Groupe :

Code 2

```

1  #include <iostream>
2  using namespace std;
3
4  class exceptA {};
5  class exceptB : public exceptA {};
6  class exceptC : public exceptB {};
7
8  void f() {    throw exceptB(); }
9
10 int main()
11 {
12     try
13     {
14         f();
15     }
16     catch(exceptA& e)
17     {
18         cout << "I_caught_an_A" << endl;
19     }
20     catch(exceptB& e)
21     {
22         cout << "I_caught_a_B" << endl;
23     }
24     catch(exceptC& e)
25     {
26         cout << "I_caught_a_C" << endl;
27     }
28     catch (...)
29     {
30         cout << "I_caught_something" << endl;
31     }
32 }
```

Affichage souhaité du Code 2

1 I caught a B

Affichage produit par le Code 2

1 I caught an A

Matricule :

Nom :

Prénom :

Groupe :

Code 3

```

1 #include <iostream>
2 #include <vector>
3 struct A {
4     int i;
5     A(int i) : i(i) {}
6     void print() { std::cout << i << "_"; }
7 };
8 int main() {
9     std::vector<A> v(5);
10    for(int j = 0; j < v.size(); j++)
11        v[j].i = j;
12    for(A a : v)
13        a.print();
14 }
```

Affichage souhaité du Code 3

```

1 0 1 2 3 4
```

Affichage produit par le Code 3

```

1 Erreur de compilation
```

Code 4

```

1 #include <iostream>
2 struct A {
3     int i;
4     A operator +(int i) { std::cout << "For_Maccrage_we_march" << std::endl; }
5     A operator +(A a) { std::cout << "and_we_shall_know_no_fear" << std::endl; }
6 };
7 int main() {
8     int i1; A a1; A a2;
9     i1 + a1 + a2; //vous ne pouvez pas changer cette ligne
10 }
```

Affichage souhaité du Code 4

```

1 For Maccrage we march
2 and we shall know no fear
```

Affichage produit par le Code 4

```

1 Erreur de compilation
```

Matricule :

Nom :

Prénom :

Groupe :

Code 5

```

1  #include <iostream>
2  class A {};
3  struct
4  {
5      A * a;
6      B() : a(new A()) {}
7      ~B() { delete a; }
8      void print() { cout << "Get_to_the_choppaaa" << endl; }
9  };
10
11 void f(B b) { std::cout << "You_have_no_respect_for_logic" << std::endl; }
12
13 int main()
14 {
15     B b;
16     f(b);
17     b.print();
18
19     B bb;
20     bb = b;
21     bb.print();
22
23     bb = bb;
24     bb.print();
25 }
```

Affichage souhaité du Code 5

```

1  You have no respect for logic
2  Get to the choppaaa
3  Get to the choppaaa
4  Get to the choppaaa
```

Affichage produit par le Code 5

```

1  Erreur de segmentation
```

Matricule :

Nom :

Prénom :

Groupe :

Code 6

```

1 #include <iostream>
2 template<class T> struct Poney {
3     T & t; int & i;
4     Poney(T & t) { this->t = t; }
5     Poney(int & i) { this->i = i; }
6     void print() { std::cout << t << " " << i << std::endl;}
7 };
8 int main() {
9     Poney<int> b(2); b.t = 3;
10    b.print();
11 }

```

Affichage souhaité du Code 6

```

1 3 2

```

Affichage produit par le Code 6

```

1 Erreur de compilation //il y a potentiellement deux erreurs dans ce code

```

Code 7

```

1 #include <iostream>
2 template<class T> struct MyVector {
3     std::vector<T> v;
4     MyVector(std::vector<T>& v) : v(v) {}
5 };
6 template<class Container, class Fct> void forall (Container c, Fct f) {
7     for(auto e : c)
8         f(e);
9 }
10 int main() {
11     std::vector<int> v = {0, 1, 2, 3, 4, 5};
12     MyVector<int> myv(v);
13     forall(myv, [](int i) { std::cout << i << " "; });
14 }

```

Affichage souhaité du Code 7

```

1 0 1 2 3 4 5

```

Affichage produit par le Code 7

```

1 Erreur de compilation

```

Code 8

```

1  #include <iostream>
2  #include <list>
3  #include <algorithm>
4
5  class MyList
6  {
7      std::list<int> l;
8      public:
9          MyList(int i = 0) : l(std::list<int>(i)) {}
10         void add(int i) { l.push_back(i); }
11         void print()
12         {
13             for(int i : l)
14                 std::cout << i << " ";
15             std::cout << std::endl;
16         }
17     };
18
19     int main()
20     {
21         std::list<MyList<int>> list(5);
22         for(auto it = list.begin(), int n = 4; it != list.end(); it++, n--)
23         {
24             MyList<int> l;
25             for(int i = n; i >= 0; i--)
26                 l.add(i);
27             *it = l;
28         }
29         std::sort(list.begin(), list.end()); // trie les listes par taille croissante
30         for(MyList<int> l : list)
31             l.print();
32     }

```

Affichage souhaité du Code 8

```

1  0
2  0 1
3  0 1 2
4  0 1 2 3
5  0 1 2 3 4

```

Affichage produit par le Code 8

```

1  Erreur de compilation

```

Matricule :

Nom :

Prénom :

Groupe :

Code 9

```

1 #include <iostream>
2 template<int N> struct Number {
3     int n;
4     Number() : n(N) { std::cout << n << std::endl; }
5 };
6 int fib(int n) {
7     if(n == 0 || n == 1) return 1;
8     else return fib(n-1) + fib(n-2);
9 }
10 int main() {
11     Number<fib(5)> n;
12 }
```

Affichage souhaité du Code 9

1 8

Affichage produit par le Code 9

1 Erreur de compilation

Code 10

```

1 #include <iostream>
2 int f(const char tab[]) {
3     return sizeof(tab) / sizeof(*tab);
4 }
5 int main()
6 {
7     char bat[] = {1, 2, 3, 4, 5, 6, 7};
8     std::cout << (sizeof(bat) / sizeof(*bat)) << std::endl;
9     std::cout << f(bat) << std::endl;
10 }
```

Affichage souhaité du Code 10

1 7
2 7

Affichage produit par le Code 10

1 7
2 8

Matricule :

Nom :

Prénom :

Groupe :

Code 11

```

1  #include <iostream>
2  using namespace std;
3
4  struct Mere
5  {
6      void f(int n) {cout << "Mere::integer_" << n << endl; }
7      void f(char n) {cout << "Mere::character_" << n << endl; }
8  };
9
10 struct Fille : public Mere
11 {
12     void f(int n, int m)
13     {
14         cout << "Fille::integers_" << n << "_" << m << endl;
15     }
16 };
17
18 int main()
19 {
20     int n = 1;
21     char c = 'a';
22     Fille b;
23     b.f(n);
24     b.f(c);
25     b.f(n, c);
26 }

```

Affichage souhaité du Code 11

```

1  Mere::integer
2  Mere::character
3  Fille::integers

```

Affichage produit par le Code 11

```

1  Erreur de compilation

```

Matricule :

Nom :

Prénom :

Groupe :

Code 12

```

1  #include <iostream>
2  using namespace std;
3
4  class Licorne
5  {
6  public:
7      Licorne() { cout << "+_"; }
8      Licorne(const A& a) { cout << "r_"; }
9      ~Licorne() { cout << "-_"; }
10     Licorne& operator =(const Licorne& a)
11     {
12         cout << "=_" << endl;
13         return *this;
14     }
15 };
16
17 void f(Licorne a) {}
18 void g(Licorne& a) {}
19
20 int main()
21 {
22     Licorne a;
23     f(a);
24     Licorne * aa = new Licorne();
25     aa = &a;
26     g(*aa);
27 }
```

Affichage souhaité du Code 12

```

1  + r - + = - -
```

Affichage produit par le code 12

```

1  + r - + -
```

Code 13

```

1  #include <iostream>
2  struct Balle {};
3  struct Joueur
4  {
5      Joueur* cible;
6      Joueur(Joueur* j = nullptr) : cible(j) {}
7      void lance(Balle ball)
8      {
9          try
10         {
11             throw ball;
12         }
13         catch(Balle& b)
14         {
15             std::cout << "J'ai attrapé la balle" << std::endl;
16             cible->lance(b);
17         }
18     }
19 };
20
21 int main()
22 {
23     Joueur p1;
24     Joueur p2(&p1);
25     p1.cible = &p2;
26
27     Balle b;
28     p1.lance(b);
29 }

```

Affichage souhaité du Code 13

```

1  J'ai attrapé la balle
2  J'ai attrapé la balle

```

Affichage produit par le code 13

```

1  Erreur d'exécution

```

Question 2. Expliquez la différence entre la ligature statique et la ligature dynamique des liens, et la façon de mettre en œuvre ces différents concepts.

/10

Question 3. Expliquez les différences principales entre le mécanisme des generics en Java et les templates en C++.

/10

Matricule : Nom : Prénom : Groupe :

Question 4. Énoncez cinq concepts non liés à la programmation orienté objet qui existent en C++ et pas en C.

/5

Question 5. Décrivez comment mettre en œuvre des conversions implicites définies par l'utilisateur dans les cas suivants

/10

- d'un type de base vers un autre type de base ;
- d'un type de base vers une classe ;
- d'une classe vers un type de base ;
- d'une classe vers une autre classe.

Question 6. Illustrez les différences de fonctionnement en matière d'allocation mémoire entre `std::vector` et `std::list` dans le cas d'ajout d'éléments à une position arbitraire du conteneur.

/10