Note/18pt syntaxe juste est ob Structures, tableau) classes, objets intervalle est soit vide, soit défini par daux bornes inf of sup get, dans le contexte de cet exercice, sont des valeurs entières. Par convention, fout intervalle Completez les déclarations et définitions suivantes. permettant de gerer de intervalles en langage C.

ayant une borne int plus grande que sa borne sup sera coasideré comme vida.

struct intervalla

0.5pt

Déclarez ici la structure intervalle.

Donnez ici la définition complète d'une fonction min acceptent deux entiers en paramètres et retournant le plus petit des deux.

int min (int a, int b) return a 4 b ? a : b;

ou if(a<b) return a; else return b

0.75pt

void printIntervalle(intervalle* w)

if (w -> inf >= w -> sup) else ("[]"); // cost<<"[]";

tervalle est vide. ok avec cout<<

Affiche un intervalle sous la forme [min, max] ou [] si l'in-

print ("[" do " d]" w sinf , w sup);

Retourne vrai si l'intersection des deux intervalles n'est pas vide, faux sinon.

Réalisez un code concis en utilisant la fonction min définie précédemment et une fonction max dont on supposera l'existence.

int testIntersect(intervalle* a, intervalle* b)

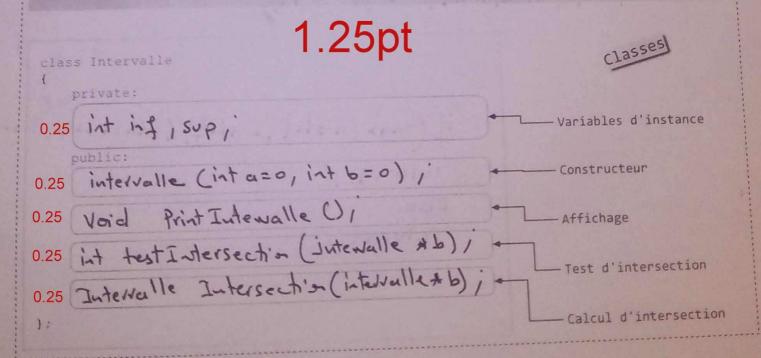
if (max (a > inf, b > inf) (min (a > sup, b > sup) return 1 1.5pt Rise return 01

Structures

ALC

```
intervalle intersection(intervalle* a, intervalle* b)
   Retourne l'intervalle (
   eventuellement vide
                               intervalle i , 0.25 avec return i
   résultat de l'intersec-
                               if (test Intervection (a,b)) 0.25
   tion de deux intervalles.
   Utilise la fonction tes-
   tIntersect presentee pre-
                                    i.inf = max (a > inf, b > inf); 0.25
                                     i.sup = min (a>sup, b>sup) 10.25
                  1pt
                                 return i;
                                                          Écrivez ici le code per-
   void testIntervalle()
                                                          mettant de créer les
   intervalle a, b, c, ab, be, ac;
                                                          intervalles [1,3], [2,4],
                                                          et [5,6], de les affi-
0.25 ainf=1; a.sup=3; PrintIntervalle(Ba);
                                                          cher, et d'afficher leurs
                                                          intersections.
0.25 d. sup= $; b.inj = 2; 11
                                         (36),
                                          (30)
0.25 c.in = 5 ; C. Sup = 6;
                                                     - [1/3] 1.5 pt
0.25 ab = laters echion (89,8b) ; 11
                                         (346);
              11 (36,30); 11
                                          (9bc),
                                                  · -> [216]
                    (3a, 9c); 11
                                           (3au);
0.25 ac =
```

Vous devez maintenant réaliser une classe Intervalle en langage C++ avec les mêmes fonctionnalités que celles décrites précédemment. Seuls la déclaration de la classe et les codes complets du constructeur et de la méthode d'affichage sont demandés.



parinition complète du construc-0.25 intervalle : intervalle (int a, int b teur de la classe Intervalle 0.5pt Classes void intervalle :: Printintervalle () 0.25 0.25 if (ht) = sup) costec " []" keendl; Définition complète de la méthode d'affichage d'un intervalle. 0.25 costes"[" (" (sup (")" 0.75pt

```
On déclare de la manière suivante une classe ListeInter représentant une liste d'interval-
les. Complétez les définitions demandées.
                                       n est le nombre d'intervalles actuellement enregis-
class ListeInter
                                       trés dans la liste et capacite est le nombre maximum
                                       d'intervalles pouvant y être stockés.
    private:
    Intervalle* tab;
    int n;
    int capacite;
                                       Le constructeur crée une liste vide, mais ayant une
                                       capacité de stockage précisée en paramètre.
    public:
    ListeInter(int capa);
    void add (Intervalle i);
                                        La méthode add ajoute un intervalle à la liste. La
    void retireVides();
                                        méthode retireVides retire de la liste tous les
                                        intervalles vides.
ListeInter::ListeInter(int capa)
    n=0 ; 0.25
    capacite = Capa, 0.25
   tab = new intervalle Tcapa); 0.5
                                                                  0.75pt
```

2pt

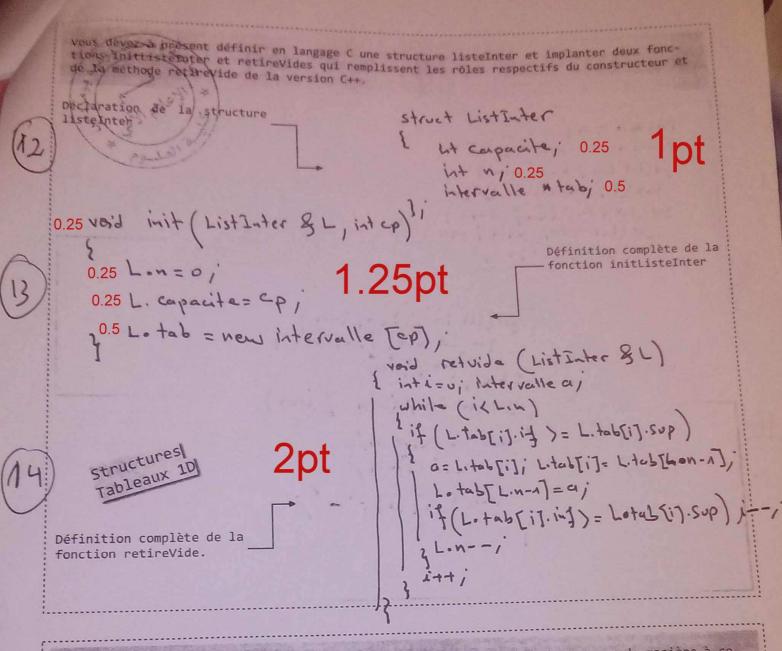
Adoptez un algorithme simple et efficace, de complexité en temps linéaire, qui peut modifier l'ardre des intervalles restants dans la liste.

Il faut utiliser ici une seule boucle

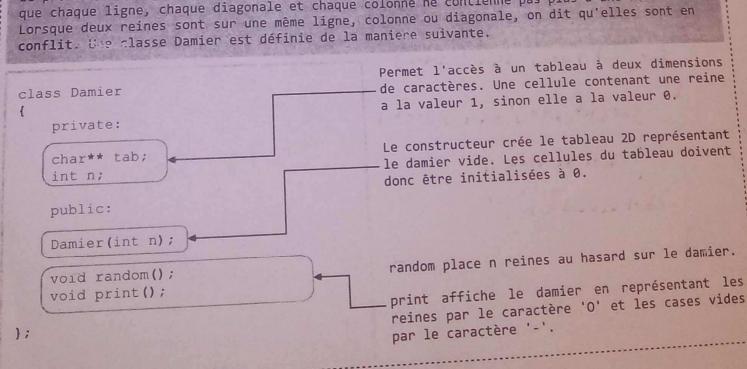
et un passage

si le code est juste mais avec 2 boucles il faut compter 1pt au lieu de 2pt

void ListeInter::retireVides() int i= o; intervalle a; while (ich) if (tab [i]. inf >= tab[i]. sup) a = tab[i]; tab[i] = tab[n-1]; tab[n-1] = 0 if (tab[i].inf >= tab[i].sup) i--, n -- /



Le problème des reines consiste à placer n reines sur un damier de n par n de manière à ce que chaque ligne, chaque diagonale et chaque colonne ne contienne pas plus d'une reine.



Complétez les définitions des méthodes indiquées.

Damier::Damier(int n)

this in = n i 0.25

tab = new char x [n] i 0.25

fa (int i=0 i kn i int)

tab [i] = new char [n] i 0.25

sar (int i=0 i kn i int)

Sar (int i=0 i kn i int)

void Damier::random (

for (int i=0 i kn i int)

tab [i] = new char [n] i 0.25

ab [int i=0 i kn i int)

tab [i] = new char [n] i 0.25

sar (int i=0 i kn i int)

tab [i] = new char [n] i 0.25

tab [i] =

1pt

Crée et initialise à 0 un damier de n par n.

Remet le damier à 0 puis place n reines aléatoirement dans des cases distinctes.
Pour mémoire, rand() % n retourne un entier aléatoire compris entre 0 et n-1 inclus.

Void Damier::random()

Gov (int i=0; ien; ita)

gov (int deo; ben; ben)

tab[i]=0; 0.25

int x, y;

gov (int i=0; i < 4 ; int)

X=rad () % N; 0.25

Y=rand () % N; 0.25

Tand () % N; 0.25

0.75pt

void Damier::print()

for (int i=0; i<n; i++)

gor (int i=0; i<n; i++)

[cont << tal ; 0.25

Cont << e-dl; 0.25

0.5pt

Affiche le damier sou forme de caractères.

5/5