


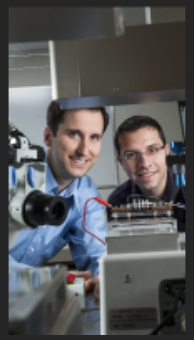




POLYTECHNIQUE MONTRÉAL

LE GÉNIE EN PREMIÈRE CLASSE

LOG3210
Éléments de langage et compilateurs

Révision

Semaine 13

EXERCICE RÉCAPITULATIF

```
main (): // Q1
   Obj a = new Obj(64);
   a.b = new Obj(16);
   Obj c = new Obj(32);
   Obj d = new Obj(12);
   bar(a);
   gc();
   foo(d);

bar (x, y): // Q2
   y = null;
   x.b = new Obj(128);
   gc();
   y = new Obj(8);
   Obj e = new Obj(8);
```

`Obj(`*size*`)` où *size* correspond à la taille de l'objet.

Q1: Dessinez l'arbre d'activation pour l'exécution de ce programme.

Q2: Donnez l'état de la pile suite à l'appel à *bar()*. Complétez les différents champs des enregistrements d'activation.

EXERCICE RÉCAPITULATIF

```
main ():
   Obj a = new Obj(64);
   a.b = new Obj(16);
   Obj c = new Obj(32);
   Obj d = new Obj(12);
   bar(a);
   gc(); // Q5
   a = new Obj(8);
   gc(); // Q6
bar (x, y):
   y = null;
   x.b = new Obj(128);
   gc(); // Q3
   y = new Obj(8); // Q4
   Obj e = new Obj(8);
```

Q3: Appliquez l'algorithme *Mark-and-Sweep*.

Q4: Allouez *y* selon la stratégie *First-Fit*, *Best-Fit* et *Next-Fit*.

Q5: Appliquez l'algorithme *Mark-and-Compact*.

Q6: Appliquez l'algorithme *Copy Collector*.

EXERCICE RÉCAPITULATIF

```
main ():
  Obj a = new Obj(64);
  a.b = new Obj(16);
  Obj c = new Obj(32);
  Obj d = new Obj(12);
  bar(a);
  gc();
  d.i = 42;
  d.j = foo(d.i); // Q9
foo (x): // Q7, Q8
  a = 2 * x; y = z = 0;
  while (x > 0):
      y = x + z;
      z = x--;
  ret a + y - z;
```

Q7: Générez le code intermédiaire pour le fonction *foo*.

Q8: Générez le code machine pour la fonction *foo* avec 3 registres.

Q9: Dessinez l'état de la pile juste avant le retour à la fonction *main*. Complétez les différents champs des enregistrements d'activation.

EXERCICE RÉCAPITULATIF

```
main ():
   Obj a = new Obj(64);
   a.b = new Obj(16);
   Obj c = new Obj(32);
   Obj d = new Obj(12);
   bar(a);
   gc();
   d.i = 42;
   d.j = foo(d.i);
foo (x): // Q10, Q11, Q12
   a = 2 * x; y = z = 0;
   while (x > 0):
       y = x + z;
       z = x--;
   ret a + y - y;
```

Q10: Construisez le DAG et utilisez les identités algébriques pour simplifier la fonction *foo*.

Q11: Y a-t-il d'autres optimisations possibles ?

Q12: Quel est l'impact sur la génération du code machine ?