Tentamen
C++-programmering

2013–04–06, 8.00–13.00

Hjälpmedel: En valfri C++-bok. OH-bilderna från föreläsningarna är *inte* tillåtna.

Du ska i dina lösningar visa att du behärskar C++ och att du kan använda C++ standardklasser. ”C-lösningar” ger inga poäng, även om de är korrekta.

Uppgifterna ger preliminärt $20 + 6 + 18 + 6 = 50$ poäng. För godkänt krävs 25 poäng (3/25, 4/33, 5/42).

---

1. En klass som beskriver en mängd av heltal har följande specifikation:

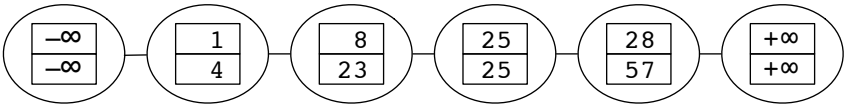
```
class IntSet {
public:
    IntSet();                     // create empty set
    bool contains(int nbr) const; // true if nbr is in the set
    void insert(int nbr);         // insert a number
};
```

   a) Implementera klassen med hjälp av STL-klassen `set`.

   b) I den aktuella tillämpningen är det vanligt att flera tal i följd lagras; till exempel kan en mängd innehålla talen 1–4, 8–23, 25, 28–57. Detta ska man utnyttja för att spara minnesutrymme.

   Implementera klassen enligt följande anvisningar:

   - Mängden ska implementeras med en länkad lista (STL-klassen `list`).
   - Elementen i listan ska vara objekt av en klass `Element` som har två `int`-medlemsvariabler `min` och `max`. `min` anger det minsta talet i en följd av tal, `max` det största talet. En lista kan till exempel innehålla elementen (1,4), (8,23), (25,25), (28,57).
   - Elementen ska vara sorterade i växande ordning (men du behöver inte söka med binärsökning).
   - Först i listan ska finnas ett element med $min = max = -\infty$, sist ett element med $min = max = +\infty$ (använd `numeric_limits<int>::min()` respektive `max()` i programmet). Dessa element underlättar eftersom varje element i listan kommer att ha både en föregångare och en efterföljare. Listan i ett tomt `IntSet` innehåller båda dessa element. Exempel:



   - Flera fall kan inträffa när man lägger in ett tal i mängden:
     - `insert(12)`: talet finns redan, ingenting ska inträffa.
     - `insert(24)`: två element ska slås ihop till ett.
     - `insert(7)`: `min`-värdet i ett existerande element ska ändras.
     - `insert(5)`: `max`-värdet i ett existerande element ska ändras.
     - `insert(6)`: ett nytt element ska läggas in på rätt plats i listan.

2. Skriv en funktion som gör att man kan skriva ut innehållet i ett `IntSet`-objekt (uppgift 1b, implementeringen med lista) med << på vanligt sätt. Om `is` är mängden i exemplet i 1b ska `cout << is` ge utskriften `1 2 3 4 8 9 ... 22 23 25 28 29 ... 56 57` (i stället för punkterna ska du skriva ut alla tal i intervallet). Du måste också göra en ändring i klassen `IntSet` — vilken?

3. Ett anagram är ett ord som man har fått fram genom att kasta om bokstäverna i ett annat ord (*admirer* är ett anagram av *married*, till exempel). Ord som är anagram av varandra kallar vi en anagramgrupp. Skriv ett program som läser en fil *dict.txt* (med ett ord per rad) och skriver ut orden som ingår i den största anagramgruppen (eller grupperna, om det finns flera som är lika stora). Exempel:

```
The largest anagram groups:
carets caster caters crates reacts recast traces
pares parse pears rapes reaps spare spear
```

4. Två klasser som beskriver personer respektive bilar som ägs av personer:

```
class Person {
public:
    Person(const string& nm) : name(nm) {}
    string getName() const { return name; }
private:
    string name; // the person's name
};

class Car {
public:
    Car(const string& n, Person* p) : nbr(n), owner(p) {}
    string getNbr() const { return nbr; }
    const Person* getOwner() const { return owner; }
private:
    string nbr;    // license number
    Person* owner; // owner, pointer so a person may own more than one car
};
```

Klasserna finns i ett bibliotek och kan *inte* modifieras. De används i följande funktion:

```
void f() {
    vector<Car> v;
    v.push_back(Car("ABC123", new Person("Johan")));
    v.push_back(Car("XYZ789", new Person("Bo")));
    v.push_back(Car("DEF456", new Person("Anna")));
    copy(v.begin(), v.end(), ostream_iterator<Car>(cout, "\n"));
}
```

Avsikten är att `copy`-anropet ska ge utskrift av innehållet i vektorn på följande form:

```
ABC123 Johan
XYZ789 Bo
DEF456 Anna
```

a) Vid kompilering får man en massa felutskrifter på `copy`-raden. Varför? Gör ett tillägg som gör att felutskrifterna försvinner och att funktionen ger rätt resultat (`copy`-raden ska vara oförändrad).

b) Funktionen innehåller en minnesläcka. Korrigera den.

c) Skriv två anrop av STL-funktionen `sort`: ett som sorterar vektorn `v` efter bilarnas registreringsnummer, ett som sorterar efter ägarnas namn. (Du får definiera hjälpfunktioner eller hjälpklasser).