Informatica III – 15/1/ 13 – info 3

Al PC

Record di attivazione A

Scrivi una funzione in **C** che dato in ingresso una stringa conta il numero di caratteri 'a' in essa. Al solito, scrivi tre versioni: una non ricorsiva, una ricorsiva senza tail recursion e una ricorsiva con tail recursion.

Specifica esattamente i parametri che passi alla procedura, il tipo di passaggio utilizzato e il loro significato.

Scrivi anche un main di esempio in cui chiami la funzione con una stringa (corta) di esempio e invocala in modo assegni il risultato ad una variabile locale al main. Non usare alcuna variabile globale.

Disegna il record di attivazione per tutte e tre le versioni fino alla massima estensione del record di attivazione. Nel caso di tail recursion, spiega quali ottimizzazioni hai adottato o potresti adottare.

Tipi opachi in C

Implementa con i tipi opachi in C un tipo che rappresenta una automobile con alcune operazioni (accelera e frena) e scrivi un piccolo esempio.

Key-Hoare

Scrivi un programma che calcola il risulatato della divisione intera tra due numeri interi, scrivi pre (meglio se pre = true) e post condizioni e dimostra la correttezza. Guarda il file di esempio nello zip fornito.

SU FOGLIO

Record di attivazione in C++

Considera questo codice C++:

```
int MAX = 20;

int f_a(int x, int* y){
     int& q = x;
     if (q > 0 ){
       return f_a(-x,y);
     }else{
       int y = 90;
       return f_b(y);
     }
}
int f_b(int a){
    return MAX + a;
}

int main(){
      int q = 90;
      cout << f_a(q,&MAX);
}
```

Quale è l'output? Scrivi il record di attivazione fino alla sua massima estensione. Puoi ottimizzare qualcosa?

C++

Considera il seguente codice?

```
class Albergo {
 private:
   int pri() { return 1; }
 public:
   int pub() { return 2; }
   virtual int vpub() { return 3; }
};

class AlbergoLusso: public Albergo {
 private:
  int pri() { return 6; }
 public:
  int vpub() { return 7; }
};

class Rifugio: private Albergo {
 public:
   int vpub() { return 5; }
};

int main() {
 1.    Albergo p;
 2.    cout << p.pri() << endl;
 3.    cout << p.pub()<< endl;
   4.  cout << p.vpub()<< endl;

 5.    Rifugio s;
 6.    cout << s.pri() << endl;
 7.    cout << s.pub()<< endl;
 8.    cout << s.vpub()<< endl;

 9.    AlbergoLusso m;
10.    cout << m.pri() << endl;
11.    cout << m.pub()<< endl;
12.    cout << m.vpub()<< endl;

13.    Rifugio * pv = &p;
14.    AlbergoLusso * pm = &p;

15.    Albergo * pp = &s;
16.    cout << pp->pri()<< endl;
17.    cout << pp->pub()<< endl;
18.    cout << pp->vpub()<< endl;

19.    Albergo * pp2 = &m;
20.    cout << pp2->pri()<< endl;
21.    cout << pp2->pub()<< endl;
22.    cout << pp2->vpub()<< endl;

23.    Albergo e1 = s;
24.    cout << e1.pri() << endl;
25.    cout << e1.pub()<< endl;
26.    cout << e1.vpub()<< endl;

27.    Albergo e2 = m;
28.    cout << e2.pri() << endl;
29.    cout << e2.pub()<< endl;
30.    cout << e2.vpub()<< endl;}
```

Quale è l'output/effetto prodotto da ogni linea numerata del main? Se contiene un errore scrivi errore, spiega l'errore e ignora la linea (ed eventuali istruzioni che dipendono da essa).

Dinamic Binding in Java

Date le seguenti dichiarazioni:

```
class Albergo{
   public int m(long d){return 5;}
}

Object oc = new Albergo();
Albergo cs = new AlbergoLusso();
```

```
class AlberoLusso  extends Albergo{
   public int m(long d){return 6;}
   public int m(int d){return 7;}
}
AlbergoLusso ss = new AlbergoLusso();
```

Quale è il valore delle seguenti tre espressioni spiegando bene (cioè anche il processo di early e late binding dove necessario) il perché (anche se le ritieni errate):

```
oc.equals(oc);
oc.equals(cs);
cs.m(2);
```

```
cs.m(2L);
ss.m(2);
ss.m(2L);
```

Java

C'è la covarianza tra tipi generici rispetto al tipo base? Ad esempio se A è sottoclasse di B, C<A> è sottoclasse di C<B>? Quale è la gerarchia dei tipi e supertipi di C<A>?

Testing – 6 crediti

Testing programmi

Dato il seguente programma:

```
foo(int x, int y){
     if( y % 6 == 0 || (x > 0 && y == -5)){
          x--;
     }
}
```

Disegna il grafo del programma e trova i casi di test per la copertura delle istruzioni, delle decisioni, delle condizioni e l'MCDC

Testing FSM

Scrivi una macchina a stati finiti con almeno 3 stati e trovane un transition tuor (anche non euleriano). Introduci un difetto e scopri se il tuo test è in grado di scoprirlo.