INFORMATICA III – prova del 7 Aprile

NOME_____

1 Macchina di Turing

Scrivere le istruzioni (stato, simbolo letto, simbolo scritto, movimento testina, nuovo stato) di una macchina di Turing deterministica con alfabeto (s0, 0, 1) che, preso in input un numero binario sul nastro, conta quanti 1 e quanti 0 ci sono nel numero e alla fine lascia sul nastro solo 1 se ci sono più 1 e lascia 0 se ci sono più zeri. Se il numero di 0 e 1 è uguale lascia il nastro vuoto. Assumi pure che il numero sia a sinistra rispetto la posizione iniziale della testina.

2 C

Il seguente programma C (ignora include mancanti) è corretto? Quali problemi ha? Mostra l'output atteso?

```
int* example(){
  x=(int *)malloc(sizeof(int));
  *x = 5;
  free(x);
  return (x);
}
void main () {
  int *B = example();
  printf("%d\n", *B);
}
```

3 Cyclone

Scrivi un esempio in ciclone di uso di @numelts(e)

4 Record di attivazione

Considera la seguente funzione:

```
void f(int d[], int n)
{
   if (n != 0)
   {
      cout << d[n - 1] << endl;
      f(d, n - 1);
   }
}
```

Con la seguente chiamata

```
int d[3] = { 3, 89, 47 };

f(d, 3);
```

Disegna con esattezza i record d'attivazione (con tutti i link tranne il Retrun address) e spiega come evolve nel tempo.

5 Passaggio parametri

Considera il seguente codice C++:

```
   void Mystery( int & a, int & b, int c ){

     a = b + c;
     b = 0;
     c = 0;
   }

  void Print(){
```

```
    int x = 0, y = 1, z = 2;

    Mystery(x, y, z);
    cout << x << " " << y << " " << z;
    Mystery(x, y, z);
    cout << x << " " << y << " " << z << endl;
}
```

1. la funzione Print Cosa stampa?
2. Se modifichiamo la dichiarazione di p passando I parametri per riferimento, mystery `(int & x, int &y, int &z){...},` quale è il valore stampato?

6 Dynamic Binding in Java

Date le seguenti dichiarazioni:

```
class Pizza{
   public boolean equals(Pizza p){
          System.out.println("P");
          return true;
}}
```

```
class PizzaMargherita extends Pizza{
   public boolean equals(PizzaMargherita
l){
          System.out.println("M");
          return true;
}}
```

dire con precisione quali di TUTTE le seguenti istruzioni sono corrette e quale l'output da esse prodotto (se c'è un errore, scrivi ERRORE, ignoralo e continua, se l'istruzione è corretta ma non ha output scrivi OK) – spiegando bene (cioè anche il processo di early e late binding dove necessario) il perchè:

	OUTPUT	MOTIVO
`Object o = new Pizza();`		
`Pizza p = new Pizza();`		
`Pizza p2 = (Pizza)o;`		
`PizzaMargherita l = new PizzaMargherita();`		
`Pizza p3 = new PizzaMargherita();`		
`p.equals(p);`		
`p.equals (o);`		
`p3.equals (p3);`		
`p3.equals(l);`		
`l.equals(p3);`		

l.equals(l);		

7 C++

Considera il seguente programma. Quale è l'aoutput?

```
#include <iostream>
using namespace std;

class base {
public:
          void Iam()   { cout << "1 - base::Iam()"   << endl; }
  virtual void Iam_v() { cout << "2 - base::Iam_v()" << endl; }
};

class drvd : public base {
public:
  virtual void Iam()   { cout << "3 - drvd::Iam()"   << endl; }
  virtual void Iam_v() { cout << "4 - drvd::Iam_v()" << endl; }
};

class drvddrvd : public drvd {
public:
  virtual void Iam()   { cout << "5 - drvddrvd::Iam()"   << endl; }
  virtual void Iam_v() { cout << "6 - drvddrvd::Iam_v()" << endl; }
};

int main() {
  base b;
  drvd d;
  drvddrvd dd;
  base *pb;
  drvd *pd;
```

b.Iam();
b.Iam_v();
d.Iam();
d.Iam_v();
dd.Iam();
dd.Iam_v() ;
pb = &b ;
pb->Iam();
pb->Iam_v();
pb = &d;
pb->Iam();
pb->Iam_v();
pb = ⅆ
pb->Iam();
pb->Iam_v();

```
  pd = &d;
  pd->Iam();
  pd->Iam_v();

  pd = &dd;
  pd->Iam();
  pd->Iam_v();
}
```

8 Semantica Assiomatica

Fai un sempio di applicazione della regola condizionale su un programma completo di pre e post condizioni a tua scelta.

SOLUZIONE

PROGRAMMA C

```
#include<stdio.h>
#include<stdlib.h>

int* example(){
  int* x=(int *)malloc(sizeof(int));
  *x = 5;
  //free(x); <<<< dangling pointer
  return (x);
}
void main () {
  int *B = example();
  printf("%d\n", *B);
}
```

CYCLONE

```
...
int * @numelts(8) foo = new {1,2,3,4,5,6,7,8};
...
```

PASSAGGIO RIFERIMENTI:

3 0 2

2 0 2

DINIMIC BINDIG

OK

OK

OK

OK

OK

P

true

P

P

P

M

PROGRAMMA C++

```
  // static bindings
```

```
  b.Iam();     // base::Iam()            1
  b.Iam_v();   // base::Iam_v()          2
  d.Iam();     // drvd::Iam()            3
  d.Iam_v();   // drvd::Iam_v()          4
  dd.Iam();    // drvddrvd::Iam()        5
  dd.Iam_v() ; // drvddrvd::Iam_v()      6

  // dynamic bindings for Iam_v()
  pb = &b ;
  pb->Iam();   // base::Iam()            1
  pb->Iam_v(); // base::Iam_v()          2

  pb = &d;
  pb->Iam();   // base::Iam()            1
  pb->Iam_v(); // drvd::Iam_v()          4

  pb = &dd;
  pb->Iam();   // base::Iam()            1
  pb->Iam_v(); // drvddrvd::Iam_v()      6

  // dynamic bindings now also for Iam()!!
  pd = &d;
  pd->Iam();   // drvd::Iam()            3
  pd->Iam_v(); // drvd::Iam_v()          4

  pd = &dd;
  pd->Iam();   // drvddrvd::Iam()        5
  pd->Iam_v(); // drvddrvd::Iam_v()      6
}
```