II Compitino Programmazione (13.03.2013)

Domande di Teoria:

1) Considerate le due seguenti funzioni e il main() che le invoca:

```
void f(int*&a) {cout<<a[1]<<endl;}
void g(int*a) {cout<<a[2]<<endl;}
main() {int X[]={1,2,3}; f(X); g(X);}
```

Secondo voi c'è differenza tra f e g? Quale?
(Non si tratta dal fatto che stamperebbero elementi diversi di X)

2) Il C++, oltre al cast che eredita dal C, possiede alcune nuove operazioni di cast che sono diverse da quelle del C. Spiegare brevemente il motivo di questo fatto.

3) Cosa stampa il seguente programma?

```
int i=10;
for (i=0; i<4; i++)
 {
  cout<<i<<endl;
  if((i+3)%2)
   continue;
  else
   break;
 }
cout<<i<<endl;
```

Risposte (NON ufficiali -> NON garantite)

1) Personalmente mi è sembrato che la funzione f abbia il parametro passato per riferimento mentre in g sia per valore.
ma...
compilando questo:

```
#include <iostream>
using namespace std;
void f(int*&a) {cout<<a[1]<<endl;}
void g(int*a) {cout<<a[2]<<endl;}
main() {int X[]={1,2,3}; f(X); g(X);}
```

ottengo questo:

In function 'int main()':
[line]5: error: invalid initialization of non-const reference of type 'int&' from a temporary of type 'int*'*
[line]3: error: in passing argument 1 of 'void f(int&)'*

2) Forse si tratta del "dynamic cast" ...

3) il trucco sta nelle istruzioni che seguono il for, nello specifico NON è (INT i=0; i<4; i++) ma semplicemente (i=0; i<4; i++), per questo motivo la variabile i usata nel blocco del for è LA STESSA dichiarata nel main con int i=10. La prima volta che si entra nel ciclo i viene azzerata.
Morale: il programma compila, o meglio quanto segue compila:

```
#include <iostream>
using namespace std;
main()
{
int i=10;
 for (i=0; i<4; i++)
  {
   cout<<i<<endl;
   if((i+3)%2)
    continue;
   else
    break;
  }
 cout<<i<<endl;}
```

e l'output è

0 1 1

o meglio

0 "endl" 1 "endl" 1 "endl"

cioè

0

1

1