Nome, cognome, matricola: _____

A

Scrivere l'output del codice nel campo indicato:

```
#include <iostream.h>        // ESERCIZIO 1
class A {
   public:
   A() {cout << "Z";}
   A & operator=(A& a) { cout << "T"; return *this; }
   A (A &a) { cout << "O"; }
} ;
int main(){
  A a;
  A *b;
  A c=a;
  *b=c;
  return 0;
}                  //OUTPUT: _____
```

```
#include <iostream.h>         // ESERCIZIO 2
void rimescola(int& k, int m, int* n) {
k = m; m = *n; *n = k; n[-1] = k;
}

int main(){
   int vet[] = {5,4,3,2,1};
   rimescola(vet[0],vet[2],vet+4);
  for (int i=0;i<5;i++) cout<<vet[i];
  return 0;
}

//OUTPUT: _____
```

```
#include <iostream.h>        // ESERCIZIO 3
class A {
   public:
   A() {cout << "Z";}
   void say() {cout << "W";}
} ;

int main(){
  A a[3];
  a[1].say();
  return 0;
}                  //OUTPUT: _____
```

```
        // ESERCIZIO 4
class A {
   public A() {System.out.println( "Z");}
   public void say() { System.out.println( "W");}

   public static void main(String a[]){
     A a[]=new A[3];
     a[1]=new A();
     a[1].say();
     return 0;
   }
}
```

Continua sul retro =>

A

Scrivere l'output del codice nel campo indicato:

```
#include <iostream.h>          // ESERCIZIO 5
class A {
   public:
   A() {cout << "X";}
   void say() {cout << "Y";}
   ~A() {cout << "Z";}
} ;
class B:public A{
   public:
   B() {cout << "A";}
   void say() {cout << "B";}
   ~B() {cout << "C";}
} ;
int main(){
  A *a=new B; a->say(); delete a;
return 0; }          //OUTPUT: _____
```

```
#include <iostream.h>          // ESERCIZIO 6
class A {
   public:
   A() {cout << "X";}
   void say() {cout << "Y";}
   ~A() {cout << "Z";}
} ;
class B: virtual public A{
   public:
   B() {cout << "A";}
   void say() {cout << "B";}
   ~B() {cout << "C";}
} ;
int main(){
  *a=new B; a->say(); delete a;
return 0;}           //OUTPUT: _____
```

```
#include <iostream.h>          // ESERCIZIO 7
class A {
   public:
   A() {cout << "X";}
   void say() {cout << "Y";}
   ~A() {cout << "Z";}
} ;
class B:public A{
   public:
   B() {cout << "A";}
   virtual void say() {cout << "B";}
   virtual ~B() {cout << "C";}
} ;
int main(){
  A *a=new B; a->say(); delete a;
   return 0;     //OUTPUT: _____
}
```

```
#include <iostream.h>          // ESERCIZIO 8
class A {
   public:
   A() {cout << "X";}
   virtual void say() {cout << "Y";}
   virtual ~A() {cout << "Z";}
} ;
class B:public A{
   public:
   B() {cout << "A";}
   void say() {cout << "B";}
   ~B() {cout << "C";}
} ;
int main(){
  A *a=new B; a->say(); delete a;
   return 0;     //OUTPUT: _____
}
```

Continua sul retro =>