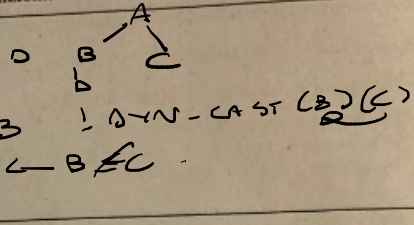


Siano A, B, C e D quattro diverse classi polimorfe. Si considerino le seguenti definizioni.

```
template <class D, class B>
D* funD(B* p) { return dynamic_cast<D>(&*p); }

main() {
    C c; funA(B, &c);
    if( funA(B, new C()) == 0 ) cout << "Bjarne ";
    if( dynamic_cast<C*>(new B()) == 0 ) cout << "Stroustrup";
    A* p = funD(B, new D());
}
```



Si supponga che:

- il main() compili correttamente ed esegua senza provocare errori a run-time;
- l'esecuzione del main() provochi in output su cout la stampa Bjarne Stroustrup.

In tali ipotesi, per ognuna delle relazioni di sottotipo $T_1 \leq T_2$ nelle seguenti tabelle segnare con una croce l'entrata

- "Vero" per indicare che T_1 sicuramente è sottotipo di T_2 ;
- "Falso" per indicare che T_1 sicuramente non è sottotipo di T_2 ;
- "Possibile" altrimenti, ovvero se non valgono né (a) né (b).

	Vero	Falso	Possibile
$A \leq B$		X	
$A \leq C$		X	
$A \leq D$		X	
$B \leq A$	X		
$B \leq C$		X	
$B \leq D$			X

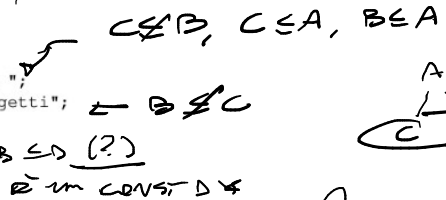
	Vero	Falso	Possibile
$C \leq A$	X		
$C \leq B$		X	
$C \leq D$			X
$D \leq A$	X		
$D \leq B$			X
$D \leq C$		X	

Esercizio 1

Siano A, B, C, D e E cinque diverse classi polimorfe. Si considerino le seguenti definizioni.

```
template <class B, class A>
B* funB(A* r) { return dynamic_cast<B>(&*r); }

int main() {
    B b; C c; D d; E e;
    if( funA(B, &c) == nullptr ) cout << "Programmazione ";
    if( dynamic_cast<C*>(&b) == nullptr ) cout << "ad Oggetti";
    const A* p = funD(B, &d);
    const D* = funE(B, &d);
    C* = e;
}
```



Si supponga che:

- il main() compili correttamente ed esegua senza provocare errori a run-time o undefined behaviour;
- l'esecuzione del main() provochi in output su cout la stampa Programmazione ad Oggetti.

In tali ipotesi, per ognuna delle relazioni di sottotipo $T_1 \leq T_2$ nella seguente tabella da ricopiare nel foglio scrivere nella corrispondente cella:

- "VERO" per indicare che T_1 sicuramente è sottotipo di T_2 ;
- "FALSO" per indicare che T_1 sicuramente non è sottotipo di T_2 ;
- "POSSIBILE" altrimenti, ovvero se non valgono né (a) né (b).

$A \leq C$	$A \leq D$	$B \leq A$	$B \leq C$	$B \leq D$	$C \leq D$	$E \leq A$	$E \leq B$	$E \leq C$	$E \leq D$
F	F	V	P	F	P	V	V	V	

Esercizio Tipi

Siano A, B, C e D classi polimorfe distinte. Si considerino le seguenti definizioni.

```
template <class X, class Y>
X* fun(X* p) { return dynamic_cast<Y*>(p); }

int main() {
    D d; funA(B, &d);
    if( funA(B, new D()) == 0 ) cout << "Bjarne ";
    if( dynamic_cast<D*>(new B()) == 0 ) cout << "Stroustrup";
    A* p = funC(B, new C());
}
```

Si supponga che:

- il main() compili correttamente ed esegua senza provocare undefined behaviour;
- l'esecuzione del main() provochi in output su cout la stampa Bjarne Stroustrup.

In tali ipotesi, per ognuna delle relazioni di sottotipo $T_1 \leq T_2$ nelle seguenti tabelle segnare con una croce l'entrata

- "Vero" per indicare che T_1 sicuramente è sottotipo di T_2 ;
- "Falso" per indicare che T_1 sicuramente non è sottotipo di T_2 ;
- "Possibile" altrimenti, ovvero se non valgono né (a) né (b).