

Assumere che le variabili `si`, `uc`, `d`, `cf`, `li`, `arr` abbiano tipo, rispettivamente, `signed int`, `unsigned char`, `double`, `const float`, `long int`, `int[50]`. Per ognuno degli accoppiamenti argomento/parametro definiti di seguito, indicare (se esiste) la categoria della corrispondente conversione implicita:

Indice	Argomento	Parametro formale	Risposta	Nota
1	si	float	P	<code>signed int</code> viene promosso a <code>float</code> .
2	(si + d)	double	E	<code>(si + d)</code> è di tipo <code>double</code> a causa della conversione standard.
3	4.0f	const double&	P	<code>float</code> viene promosso a <code>double</code> .
4	0	const unsigned char*	Errore	<code>0</code> non può essere convertito a <code>const unsigned char*</code> .
5	"hello"	const char*	E	Corrispondenza esatta.
6	"hello"	std::string	U	Conversione definita dall'utente tramite il costruttore <code>std::string</code> .
7	uc	signed int	P	<code>unsigned char</code> viene promosso a <code>signed int</code> .
8	cf	double	P	<code>const float</code> viene promosso a <code>double</code> .
9	li	int	P	<code>long int</code> viene promosso a <code>int</code> (potrebbe essere non sicuro).
10	arr	const int*	E	Gli array di <code>int</code> possono essere convertiti in puntatori costanti a <code>int</code> .

(Conversioni implicite) Le conversioni implicite del C++ sono distinte nelle seguenti categorie: corrispondenze esatte (E), promozioni (P), conversioni standard (S), conversioni definite dall'utente (U). Assumere che le variabili `sc`, `ui`, `f`, `ai` abbiano tipo, rispettivamente, `signed char`, `unsigned int`, `float`, `int[100]`, per ognuno degli accoppiamenti argomento/parametro definiti di seguito, indicare (se esiste) la categoria della corrispondente conversione implicita:

Indice	Argomento	Parametro formale	Risposta	Nota
1	ui	double	S	
2	(ui - f)	float	E	<code>ui</code> viene bilanciato (convertito implicitamente) a <code>float</code> , quindi l'espressione tra le () sarà di tipo <code>float</code>
3	0.3F	double	P	F specifica il tipo <code>float</code>
4	0	const float*	S	Se lo <code>0</code> è noto a tempo di compilazione viene convertito in <code>null_ptr</code>
5	"stringa"	const char*	E	
6	"stringa"	std::string	U	
7	sc	int*	Errore	
8	ui	const unsigned int&	E	
9	ai	int*	E	

Indice	Argomento	Parametro formale	Risposta
1	si	double	P
2	(uc + d)	double	E
3	4.0	const float&	P
4	0	const int*	E
5	"world"	const char*	E
6	"world"	std::string	U
7	si	unsigned int	S
8	cf	float	E
9	li	double	S
10	arr	const int(&)[50]	E

(Conversioni implicite) Le conversioni implicite del C++ sono distinte nelle seguenti categorie: corrispondenze esatte (E), promozioni (P), conversioni standard (S), conversioni definite dall'utente (U). Assumere che le variabili `sc`, `ui`, `f`, `ai` abbiano tipo, rispettivamente, `signed char`, `unsigned int`, `float`, `int[100]`, per ognuno degli accoppiamenti argomento/parametro definiti di seguito, indicare (se esiste) la categoria della corrispondente conversione implicita:

Indice	Argomento	Parametro formale	Risposta	Nota
1	sc	char	S	
2	2	long	P	
3	f	float	E	
4	"test"	const char*	E	
5	ui	const unsigned long&	S	
6	10.0	double	E	
7	sc	int	P	
8	ai	int*	E	
9	'c'	char	E	

(Conversioni implicite) Le conversioni implicite del C++ sono distinte nelle seguenti categorie: corrispondenze esatte (E), promozioni (P), conversioni standard (S), conversioni definite dall'utente (U). Assumere che le variabili `sc`, `ui`, `f`, `ai` abbiano tipo, rispettivamente, `signed char`, `unsigned int`, `float`, `int[100]`, per ognuno degli accoppiamenti argomento/parametro definiti di seguito, indicare (se esiste) la categoria della corrispondente conversione implicita:

Indice	Argomento	Parametro formale	Risposta	Nota
1	ui	signed int	S	
2	f	float	E	
3	3.0F	double	P	
4	ai	const void*	S	
5	sc	unsigned int	S	
6	"example"	const std::string&	U	
7	4	double	P	
8	ui	unsigned long	P	
9	'd'	int	P	

(Conversioni implicite) Le conversioni implicite del C++ sono distinte nelle seguenti categorie: corrispondenze esatte (E), promozioni (P), conversioni standard (S), conversioni definite dall'utente (U). Assumere che le variabili `sc`, `ui`, `f`, `ai` abbiano tipo, rispettivamente, `signed char`, `unsigned int`, `float`, `int[100]`, per ognuno degli accoppiamenti argomento/parametro definiti di seguito, indicare (se esiste) la categoria della corrispondente conversione implicita:

Indice	Argomento	Parametro formale	Risposta	Nota
1	sc	short	P	
2	5.0	long double	P	
3	ui	const unsigned int&	S	
4	f	double	P	
5	"stringa"	const std::string&	U	
6	0.1F	float	E	
7	ai	const int*	E	
8	'e'	unsigned long	S	
9	0	const void*	S	



(Conversioni implicite) Le conversioni implicite del C++ sono distinte nelle seguenti categorie: corrispondenze esatte (E), promozioni (P), conversioni standard (S), conversioni definite dall'utente (U). Assumere che le variabili `sc`, `ui`, `f`, `ai` abbiano tipo, rispettivamente, `signed char`, `unsigned int`, `float`, `int[100]`, per ognuno degli accoppiamenti argomento/parametro definiti di seguito, indicare (se esiste) la categoria della corrispondente conversione implicita:

Indice	Argomento	Parametro formale	Risposta	Nota
1	ui	long	P	
2	f	double	P	
3	0.5	int	P	
4	"world"	std::string	U	
5	ai	void*	S	
6	sc	unsigned long	S	
7	2.0F	double	P	
8	f	long double	P	
9	'b'	unsigned int	S	

(Conversioni implicite) Le conversioni implicite del C++ sono distinte nelle seguenti categorie: corrispondenze esatte (E), promozioni (P), conversioni standard (S), conversioni definite dall'utente (U). Assumere che le variabili `sc`, `ui`, `f`, `ai` abbiano tipo, rispettivamente, `signed char`, `unsigned int`, `float`, `int[100]`, per ognuno degli accoppiamenti argomento/parametro definiti di seguito, indicare (se esiste) la categoria della corrispondente conversione implicita:

Indice	Argomento	Parametro formale	Risposta	Nota
1	sc	int	P	
2	1.0	float	P	
3	1	double	P	
4	"hello"	const char*	E	
5	ai	const int*	E	
6	ui	const signed int&	S	
7	f	double	P	
8	sc	unsigned int	S	
9	'a'	int	P	