UNIVERSITÀ DEGLI STUDI DELLA CAMPANIA
LUIGI VANVITELLI
SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA
INDUSTRIALE E DELL'INFORMAZIONE

Cast in C

Prof. Salvatore Venticinque
Prof. Massimiliano Rak

Conversione di tipo

Per evitare errori occorrerebbe sempre effettuare operazioni tra variabili di stesso tipo

Tuttavia la conversione di tipo avviene quando in un'espressione abbiamo dati di tipo differente.

Esempi:

- conversione di un valore **int** in un valore **float**
- assegnazione del valore di una espressione a una variabile di tipo diverso

Operazione di cast

Il **cast** ***è una operazione di conversione di tipo***

- Il risultato di una espressione aritmetica dipende dal tipo degli operandi.
- Quando gli operandi sono diversi il risultato è determinato dalle regole di casting.
- Nel caso del linguaggio C le regole sono orientate a promuovere gli operandi verso tipi id dati più grandi (rappresentati su un maggiore numero di bit)

Può essere :

- **Implicito** (automatico)
- **Esplicito** (forzato dal programmatore)

Cast implicito – Esempio 1

Una variabile (operando) di tipo con rappresentazione più corta (di una seconda variabile operando) viene trasformata in una variabile con rappresentazione più lunga.

Esempio:

```
int a;
float f, g;
g = a + f; // a transformata in float
```

Precedenza nei casti impliciti

1)I tipi interi sono più piccoli dei tipi float

2)I tipo con segno sono più piccoli dei tipi unsigned

3)I tipi short sono più piccoli dei tipi long

4)double>float>long>int>short>char

Cast implicito – Esempio 2

```
#include<stdio.h>
 int main(void)
{
    int a=5,b=8;
    float c = 0;

    c = a/b;
    printf("\n [%f] \n",c);

}
```

Quanto vale c?

Cast implicito – Esempio 2

- Il compilatore sa che 'a' e 'b' sono interi
- traduce la divisione intera tra a e b il cui risultato verrà prima memorizzato in una variabile temporanea intera
- Tale valore intero temporaneo verrà assegnato alla variabile c, nonostante essa sia un float
- L'output del programma è: [0.000000]

Il meccanismo di conversione implicito può essere molto conveniente ma anche potenzialmente pericoloso (errori a run time).

- **int/int=int**
- **float/int=float**
- **int/float=float**
- **float/float=float**

Ricapitolando … il cast implicito avviene quando …

Un dato di un tipo numerico può essere automaticamente converitito in un altro tipo di dato.

es.,

```
int k = 5, m = 4, n;
double x = 1.5, y = 2.1, z;
```

Context	Example	Explanation
Espressione aritmetica con tipi differenti	`k + x` `il risultato è 6.5 .`	Il valore di `k` è convertito in `double` prima dell'operazione.

Ricapitolando … il cast implicito avviene quando …

Context	Example	Explanation
Assegnazione con tipo destinazione `double` e espressione di tipo `int`	`z = k / m;` il risultato dell'espressione è 1.25; `z` è 1.0.	Prima viene calcolata l'espressione. Poi il risultato è convertito in `double`.
Assegnazione con tipo destinazione `int` e espressione di tipo `double`	`n = x * y;` il risultato dell'espressione è 3.15; `n` è 3.	Prima viene calcolata l'espressione. Poi il risultato è convertito in `int`.

Cast esplicito

- Questa conversione di tipo ***ha priorità maggiore*** rispetto a quella automatica. O

  Due possibili sintassi:
- (data_type) variabile
- (data_type) espressione.

Esempio:

```
int a; double b; float c,d;
a = (int) c;
b = (double)d + c;
```

Cast esplicito – Esempio

```
#include<stdio.h>
 int main(void)
{
    int a=5,b=8;
    float d = 0;

     d = (float)a/(float)b;
     printf("\n [%f] \n",d);

}
```

Quanto vale d?

Cast esplicito – Esempio

- 'a' e 'b' sono individualmente convertiti in float
- il compilatore li considera float e traduce l'operazione in una divisione tra float, conservando il valore in una variabile temporanea float
- Il valore float completo della parte decimale viene assegnato a 'c'
- L'output del programma è:[0.625000]

Altri esempi

Example 1:

```
float a = 5.25;
int b = (int)a;
```

/*Explicit cast da **float** a **int**. Il valore di b è 5*/

Example 2:

```
char c = 'A';
int x = (int)c;
```

/*Explicit cast da **char** a **int**. Il valore di x è 65: il codice ASCII di 'A'*/

I valori caratteri possono essere comparati, letti, stampati e convertiti in `int`.

Altri esempi

Example 3:

```
int x=7, y=5 ;
float z;
z=x/y;   /*Here the value of z is 1*/
```

Se vogliamo il valore esatto:

```
int x=7, y=5;
float z;
z = (float)x/(float)y; /*Il valore di z è 1.4*/
```

Example 4:

```
int x=70;
char c =(char)x; /*Il valore di c è 'F': corrispondente al codice ASCII  70*/
```

Altri esempi

Il file ***limits.h*** contiene la dichiarazione delle costanti massime rappresentabili pe l tipi supportati dal compilatore:

INT_MAX, INT_MIN, ULONG_MAX, etc.

Consideriamo la seguente situazione:

```
int a = 8000000, b = 7000000, c;
c = a * b;
```

Il risultato aritmetico è **56000000000000**
Ma un int potrebbe non contenere tale valore causando un overflow.
Lo standard C non definisce particolari regole per gestire questa condizione.

La seguente condizione consente di gestire un risultato maggiore del massimo rappresentabile con il tipo int.

```
long int c;
int a=800, b = 700;
c = (long int)a * b;
```

Cast nel passaggio dei parametri

Esempio:

```
double sqrt(double x)
int somma(int x, int y);

int a=9,b=81, d;
float x,y;

d=sqrt(c);
d =somma((int)x,(int)y);
```