

Esercizi su

Overflow

Istruzioni di scelta: scelta
multipla

Operatore *sizeof*

- Restituisce la *dimensione* di un'espressione o di un tipo
- *sizeof (espressione)*
Numero di bytes (*char*) necessari per memorizzare i possibili valori dell'espressione
- *sizeof (tipo)*
Numero di bytes (*char*) necessari per memorizzare un oggetto del tipo passato come parametro

Esercizio: dimensione int 1/2

- Scrivere un programma che stampi sullo schermo la dimensione in bytes di un oggetto di tipo *int* sulla macchina su cui è eseguito

Esercizio: dimensione int 2/2

```
#include <iostream>
```

```
int main()
```

```
{
```

```
    cout<<sizeof(int)<<endl ;
```

```
    return 0 ;
```

```
}
```

Valori possibili per il tipo *int*

- Tipicamente un oggetto di tipo *int* che sta su n bytes, ossia $8*n$ bits, può contenere valori interi nell'intervallo

$$[-2^{(8*n - 1)}, 2^{(8*n - 1)} - 1]$$

- Quindi, per esempio su 4 bytes si ha

$$[-2^{31}, 2^{31} - 1] = [-2147483648, 2147483647]$$

- Lo standard prevede la presenza di costanti o funzioni per conoscere i limiti per ogni tipo di dato

Overflow

Valori possibili per il tipo *int*

- Tipicamente un oggetto di tipo *int* che sta su n bytes, ossia $8*n$ bits, può contenere valori interi nell'intervallo

$$[-2^{(8*n - 1)}, 2^{(8*n - 1)} - 1]$$

- Quindi, per esempio su 4 bytes si ha

$$[-2^{31}, 2^{31} - 1] = [-2147483648, 2147483647]$$

- Lo standard prevede la presenza di costanti o funzioni per conoscere i limiti per ogni tipo di dato

Esercizio 1/3

- Scrivere un programma che definisce una variabile *i* di tipo *int*, la inizializza ad un valore qualsiasi e
- stampa il valore di *i*
- incrementa *i* di una unita'
- stampa il nuovo valore di *i*

Esercizio 2/3

```
#include <iostream>
```

```
int main()
```

```
{
```

```
    int i = 23 ;
```

```
    cout<<i<<endl ;
```

```
    i++ ;
```

```
    cout<<i<<endl ;
```

```
    return 0 ;
```

```
}
```

Esercizio 3/3

- Modificare il programma affinché inizializzi i al seguente valore:

2147483647

- Quale sarà l'output del programma?
- Eseguirlo per controllare

Overflow

- Il valore di una espressione è **troppo grande** (in modulo) per essere contenuto
 - nel tipo di dato del risultato, oppure
 - nell'oggetto a cui si vuole assegnare tale valore
- E' stato segnalato automaticamente qualche errore durante l'esecuzione del precedente programma?
- Lo standard non prescrive segnalazioni d'errore

Esercizio 1/2

- Scrivere un programma che, letti due numeri interi in ingresso, stampi il risultato della somma e dica se tale somma ha generato overflow oppure no
- Esempio:

*Inserisci i due numeri da sommare:
2147483647 78*

2147483647 + 78 = -2147483571

Attenzione: c'e' stato overflow

Esercizio 2/2

- Attenzione ai numeri negativi !!!
- Soluzione in: *somma_overflow.cc*

Scelta multipla

Esercizio 1/2

- Scrivere un programma che chieda all'utente di scegliere tra varie opzioni e stampi il nome dell'opzione scelta
- Esempio:
 - 1 *Opzione A*
 - 2 *Opzione B*
 - 3 *Opzione C*
 - 4 *Opzione D*

Scegli un'opzione: 2
Hai scelto l'opzione B

- Esempio:

1 Opzione A

2 Opzione B

3 Opzione C

4 Opzione D

Scegli un opzione: -1

Scelta non valida

- Soluzione in: *primo_menu.cc*

Esercizio 1/2

- Scrivere un programma che chieda all'utente di scegliere tra varie opzioni e stampi il nome dell'opzione scelta
- Esempio:
 - 1 Opzione A*
 - 3 Opzione B*
 - 4 Opzione C*
 - 5 Opzione C*
 - 6 Opzione D*

Scegli un'opzione: 4
Hai scelto l'opzione C

- Esempio:

1 Opzione A

3 Opzione B

4 Opzione C

5 Opzione C

6 Opzione D

Scegli un'opzione: 2

Scelta non valida

- Soluzione in: *menu_multiplo.c*

Esercizio: calcolatrice

- Testo e soluzione in *calcolatrice.cc*