



UNIVERSITÀ
DI TRENTO

Dipartimento di Ingegneria e
Scienza dell'Informazione
DISI - Trento

Programmazione 1

07 - Esercitazione

Giovanni De Toni

giovanni.detoni@unitn.it

Attenzione

La presente esercitazione verrà trasmessa via Zoom. Essa verrà anche registrata e successivamente messa a disposizione degli studenti dell'Università degli Studi di Trento. Per gli utenti connessi attraverso Zoom, in caso non desideriate per qualunque motivo essere registrati, siete pregati di effettuare la disconnessione ora. La lezione sarà comunque visionabile in modo asincrono.

Anno Accademico 2021/2022

Nelle puntate precedenti

- Istruzione iterativa `for`

```
for (init; exp; agg) {  
    istruzioni  
}
```

```
for (int i=0; i<100; i++)  
{  
    cout << i << endl;  
}
```

Nelle puntate precedenti

- **Istruzione di salto** `break`, `return`, `continue`

```
for (int i=1; i<100; i++)  
{  
    if ( i%2 != 0 )  
        break; // return 0;  
    cout << i << endl;  
}
```

```
int i=0;  
while(i < 100) {  
    if ( i%2 != 0 )  
        continue;  
    else  
        cout << i << endl;  
    i++;  
}
```

00 - Let's begin!

- Tipo derivato `referimento-a (&)`
 - Consente di dare **nomi multipli** ad una variabile (in realtà, **ogni espressione dotata di indirizzo**)
 - Se la variabile originale viene modificata, allora anche il suo “alias” verrà modificato;
 - Ovviamente, se anche l’alias viene modificato allora anche la variabile originale subirà dei cambiamenti;

00 - Let's begin!

- Tipo derivato riferimento-a (&)

```
int x = 5;  
int & y = x;
```

```
x++;          // 6  
cout << y << endl;    // 6  
y++;          // 7  
cout << x << endl;    // 7
```

```
int k = 20;  
y = k;  
cout << x << endl;    // 20
```

00 - Let's begin!

- Tipo derivato `puntatore-a (*)`
 - Un puntatore contiene l'**indirizzo** di un altro oggetto precedentemente creato;
 - Ad un puntatore viene riservato uno spazio di memoria sempre uguale (atto a contenere l'indirizzo), però **non viene automaticamente riservato lo spazio per l'oggetto puntato!**
 - Per ottenere il valore dell'oggetto puntato devo utilizzare l'**operatore di dereference ***

00 - Let's begin!

- Tipo derivato puntatore-a (*)

```
int x = 5;
int * y = &x; // L'operatore & da l'indirizzo di memoria

x++;          // 6
cout << y << " " << *y << endl;    // 6
(*y)++;       // 7
cout << x << endl;    // 7

int k = 20;
y = &k;
cout << x << endl;    // 7
```

00 - Let's begin!



```
int X = 3
```



```
int &y = X
```


00 - Let's begin!

“Achille! Vieni qui!”



```
int X = 3
```

“Nerone! Vieni qui!”



```
int &y = X
```

00 - Let's begin!



```
int X = 3
```

00 - Let's begin!



```
int *y = &X
```

```
int X = 3
```

1 - Indovina l'output!

Quale è l'output dei seguenti tre programmi?

```
1.  #include <iostream>
2.  using namespace std;
3.
4.  int main()
5.  {
6.      int a = 32;
7.      int * ptr = &a;
8.
9.      int b = 14;
10.     int & ref = b;
11.
12.     a += b;
13.     *ptr += b;
14.
15.     cout << a << ", " << b << endl;
16.
17.     return 0;
18. }
```

```
1.  #include <iostream>
2.  using namespace std;
3.
4.  int main()
5.  {
6.      int i = 20;
7.      int * ptr = &i;
8.      (*ptr)++;
9.
10.     int j = 15;
11.     ptr = &j;
12.     (*ptr)++;
13.
14.     cout << i << endl;
15.
16.     return 0;
17. }
```

```
1.  #include <iostream>
2.  using namespace std;
3.
4.  int main()
5.  {
6.      int x = 5;
7.      int * y = &x;
8.      int ** k = &y;
9.
10.     cout << x << " " << &x << endl;
11.     cout << (*y) << " " << &y << endl;
12.     cout << (*k) << " " << &k << endl;
13.     cout << *(*k) << endl;
14.
15.     return 0;
16. }
```

2 - ASCII: La vendetta

Scrivere un programma che stampi a video il valore ASCII di tutte le lettere dell'alfabeto minuscole e maiuscole.

3 - ASCII: Il ritorno

Scrivere un programma che stampi a video il carattere corrispondente ad i valori ASCII compresi tra un intervallo a e b definito dall'utente in input.

Controllare anche che l'intervallo inserito dall'utente sia valido, altrimenti terminare il programma.

4 - Serie π (Parte 2)

Scrivere un programma che calcoli il risultato della seguente serie che approssima il valore pi greco. Il valore di approssimazione di questa serie N, viene dato dall'utente in input.

$$\sum_{i=1}^N \frac{1}{i^2} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots + \frac{1}{N^2} \qquad \sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{\pi^2}{6}$$

4 - Serie π (Parte 2)

Scrivere un programma che calcoli il risultato della seguente serie che approssima il valore pi greco. Il valore di approssimazione di questa serie N, viene dato dall'utente in input.

$$\sum_{i=1}^N \frac{1}{i^2} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \cdots + \frac{1}{N^2} \qquad \sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{\pi^2}{6}$$

Il calcolo della serie deve terminare quando, la differenza in valore assoluto tra l'i-esimo valore calcolato e il valore calcolato all'iterazione precedente (i-1) supera il coefficiente inserito dall'utente

$$|\pi_{(i-1)} - \pi_{(i)}| < N$$

4 - Serie π (Parte 2)

Scrivere un programma che calcoli il risultato della seguente serie che approssima il valore pi greco. Il valore di approssimazione di questa serie N, viene dato dall'utente in input.

$$\sum_{i=1}^N \frac{1}{i^2} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \cdots + \frac{1}{N^2} \qquad \sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{\pi^2}{6}$$

```
#include <cmath>
abs(-10 + 1);
```

Per maggiori informazioni:

<https://en.cppreference.com/w/cpp/numeric/math/abs>

5 - Somma tra primi

**Scrivere un programma che, dato in input un numero dall'utente, ritorni a video se questo numero possa essere espresso come la somma di due numeri primi.
Stampare anche i corrispondenti numeri primi a video.**

Input: 22 => Output: 17 + 5