

Un `Percorso` è formato da torrette numerate a partire da 1. Nel percorso transitano persone individuate dal nome. Ciascuna torretta può essere libera o occupata da una persona. Non è possibile inserire persone con lo stesso nome nel percorso.

Implementare le seguenti operazioni che possono essere effettuate su un `Percorso`:

✓ **`Percorso p`**

Costruttore di default, che inizializza un percorso `p`. Tale percorso contiene una torretta. La torretta è libera.

✓ **`p.inserisci(s)`**

Operazione che inserisce nella prima torretta del percorso una persona di nome `s`. L'operazione fallisce se la prima torretta è occupata, oppure se nel percorso esiste già una persona di nome `s`.

✓ **`p += n`**

Operatore di somma e assegnamento, che aggiunge `n` torrette libere al percorso `p`. *Le torrette vengono inserite in fondo al percorso.*

✓ **`cout << p`**

Operatore di uscita per il tipo `Percorso`. L'uscita ha la forma seguente:

```
[1] Rossi
[2] -
[3] Bianchi
[4] -
[5] -
```

In questo esempio, il percorso contiene 5 torrette. La seconda, la quarta e la quinta sono libere. La prima è occupata da una persona di nome *Rossi*, la terza da una persona di nome *Bianchi*.

✓ **`p.avanza(j)`**

Operazione che avanza di una posizione la persona che occupa la torretta `j`. Se la torretta `j+1` è occupata, l'operazione lascia il percorso inalterato. Se la torretta `j` è l'ultima del percorso, la persona esce dal percorso.

✓ **`Percorso p1(p)`**

Costruttore di copia, che inizializza un percorso `p1` con il valore del percorso `p`.

✓ **`~Percorso()`**

Distruttore.

Utilizzando il linguaggio C++, realizzare il tipo di dati astratto `Percorso` definito dalle precedenti specifiche. Individuare eventuali situazioni di errore, e metterne in opera un corretto trattamento. Si faccia riferimento al main riportato a pagina 2.

```

// file main.cpp

#include <iostream>
#include "compito.h"
using namespace std;

int main(){
    Percorso p;
    p.inserisci("Bianchi"); // successo
    cout << p << endl;

    p += 4;
    cout << p << endl;

    p.avanza(1);
    cout << p << endl;

    p.inserisci("Rossi"); // successo
    cout << p << endl;

    p.avanza(2);
    cout << p << endl;

    {
        cout<<"Chiamata al costruttore di copia"<<endl;
        Percorso p1(p);
        cout << p1 << endl;
    }
    cout<<"L'oggetto p1 e' stato appena distrutto"<<endl;

    return 0;
}

```

Uscita attesa

[1] **Bianchi**

[1] **Bianchi**

[2] -

[3] -

[4] -

[5] -

[1] -

[2] **Bianchi**

[3] -

[4] -

[5] -

[1] **Rossi**

[2] **Bianchi**

[3] -

[4] -

[5] -

[1] **Rossi**

[2] -

[3] **Bianchi**

[4] -

[5] -

Chiamata al costruttore di copia

[1] **Rossi**

[2] -

[3] **Bianchi**

[4] -

[5] -

L'oggetto p1 e' stato appena distrutto