

```
void scambia_int(int &a, int &b) {  
    int tmp = a;  
    a = b;  
    b = tmp;  
}  
  
int main() {  
    int x=1;  
    int y=2;  
    scambia_int(x,y);  
    cout << x << ":" << y << endl;  
}
```

C++

2:1

OUTPUT



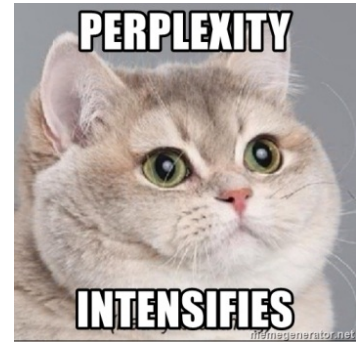
C++

```
void scambia_float(float &a, float &b) {  
    float tmp = a;  
    a = b;  
    b = tmp;  
}
```

C++

```
void scambia_string(string &a, string &b) {  
    string tmp = a;  
    a = b;  
    b = tmp;  
}
```





```
void scambia_double(double &a, double &b) {  
    double tmp = a;  
    a = b;  
    b = tmp;  
}
```

C++





```
template<typename T>  
return_type function_name(T var_name, <var_list>) {  
    code...  
}
```

C++



C++

```
template<typename T>
void scambia(T& x, T& y) { // per riferimento
    T tmp = x;
    x = y;
    y = tmp;
}

int main() {
    int a = 2; int b = 3;
    scambia(a, b);
    cout << " A = " << a << " B = " << b << endl;
    float c = 2.5; float d = 3.6;
    scambia(c, d);
    cout << " C = " << c << " D = " << d << endl;
}
```

```
A = 3 B = 2
C = 3.6 D = 2.5
```

OUTPUT



C++

```
int vettint[ ] {1,2,3,3,2,3,4,3};
int oc1 = occorrenze(vettint, 3, 8);
cout << "3 presente " << oc1 << " volte\n";

string s[ ] {"ciao", "come", "state", "come", "oggi", "come", "ieri"};
string s1 = "come";
int oc2 = occorrenze(s, s1, 7);
cout << "come presente " << oc2 << " volte\n";
```

```
3 presente 4 volte
come presente 3 volte
```

OUTPUT





```
template<typename T>
int occorrenze(T x[], T elem, int size) {    // il passaggio per array è sempre per riferimento
    int cont = 0;
    for (int i = 0; i<size; i++) {
        if (x[i] == elem) cont++;
    }
    return cont;
}
```



C++

```
template<class T>
struct nodo
{
    T info;
    nodo<T>* next;
};
```



---

C++

```
template<typename T>
void inTesta(nodo<T>* &p, T elem) {

    nodo<T>* tmp = new nodo<T>;
    tmp->info = elem;
    tmp->next = p;
    p = tmp;
}
```

---

C++

```
template<typename T>
void leggi(nodo<T>* p) {
    while(p) {
        cout << p->info << " ";
        p = p->next;
    }
}
```



C++

```
nodo<int>* intptr;  
nodo<string>* strptr;  
  
int main()  
{  
    // inizializzazione  
    intptr = nullptr;  
    inTesta(intptr, 10);  
    inTesta(intptr, 20);  
    leggi(intptr);  
  
    strptr = nullptr;  
    inTesta(strptr, (string)"molto");  
    inTesta(strptr, (string)"figo");  
    leggi(strptr);  
}
```

20 10 figo molto

OUTPUT





