

Si assumano le seguenti specifiche riguardanti la libreria Qt.

- Un oggetto della classe `QString` rappresenta una stringa di caratteri Unicode. La classe `QString` fornisce un costruttore `QString(const char*)` con il seguente comportamento: `QString(str)` costruisce una `QString` inizializzata con la stringa ASCII `str`.
- La classe `QPaintDevice` è la classe base di tutti gli oggetti che possono essere "dipinti" sullo schermo.
 - La classe `QPaintDevice` è polimorfa.
 - La classe `QPaintDevice` rende disponibile un metodo `int height() const` con il seguente comportamento: `pd.height()` ritorna l'altezza in pixel del `QPaintDevice` `pd`. È inoltre disponibile un metodo `int width() const` con analogo comportamento per la larghezza.
- `QWidget` è una sottoclasse di `QPaintDevice` i cui oggetti rappresentano delle componenti di una interfaccia grafica Qt.
 - La classe `QWidget` rende disponibile un metodo `bool hasFocus() const` con il seguente comportamento: `w.hasFocus()` ritorna `true` quando la componente `w` detiene il keyboard focus.
 - La classe `QWidget` rende disponibile un metodo `void clearFocus()` con il seguente comportamento: `w.clearFocus()` toglie il keyboard focus alla `QWidget` `w`.
- `QAbstractButton` è derivata direttamente da `QWidget` ed è la classe base astratta dei widget pulsante.
 - La classe `QAbstractButton` rende disponibile un metodo `void setText(const QString&) const` con il seguente comportamento: `b.setText(s)` setta l'etichetta testuale del `QAbstractButton` `b` alla stringa `s`.

Definire una funzione

```
vector<QAbstractButton*> fun(const vector<QPaintDevice*>& v)
```

con il seguente comportamento: in ogni invocazione `fun(v)`:

1. per ogni puntatore `p` contenuto nel vector `v`:
 - se `p` punta ad un oggetto che è un `QWidget` con altezza o larghezza > 50 pixel allora lancia una eccezione di tipo `QString` che rappresenta la stringa "TooBig";
 - se `p` punta ad un oggetto `obj` che è un `QWidget` avente sia l'altezza che la larghezza ≤ 50 pixel e che detiene il keyboard focus allora toglie il keyboard focus a `obj`;
 - se `p` punta ad un oggetto `obj` che è un `QAbstractButton` allora setta l'etichetta testuale di `obj` alla stringa "Pulsante".
2. l'invocazione `fun(v)` deve ritornare un vector contenente **tutti e soli** i puntatori `p` contenuti nel vector `v` che puntano ad un oggetto che è un `QAbstractButton`.

```
std::vector<QAbstractButton*> fun(const vector<QPaintDevice*>& v){
    std::vector<QAbstractButton*> vec;
    for(auto p = v.begin(); p != v.end(); ++p){
        QWidget* qw = dynamic_cast<QWidget*>(*p);
        if(qw && (qw->height() > 50 || qw->width() > 50)){
            throw QString("TooBig");
        }
        if(qw && (qw->height() <= 50 || qw->width() <= 50)
            && qw->hasFocus()){
            qw->clearFocus();
        }
        QAbstractButton* qab = dynamic_cast<QAbstractButton*>(*p);
        if(qab){
            qab->setText("Pulsante");
            vec.push_back(qab);
        }
    }
    return vec;
}
```