

Analice detenidamente el siguiente código e implemente la clase VectorBool (Vector de booleanos) de tal forma que este sea válido. La salida esperada se muestra al pie. Optimice su tiempo y no implemente nada que no tenga que ver con el funcionamiento adecuado del código propuesto o la administración consciente de la memoria. Se aclara que el VectorBool puede tener una longitud tan extensa como la memoria asignada a la aplicación lo permita.

```
VectorBool v1(3), v2(5), v3(3);

v1[0] = true; v1[1] = true; v1[2] = false;
v3[0] = false; v3[1] = true; v3[2] = false;
try{
    //Se intenta aplicar el operador & (and) elemento a elemento del vector
    v2 = v1 & v2;
} catch(length_error& ex) {
    cout<<"Error al intentar aplicar v1 & v2..."<<ex.what()<<endl;
} catch(exception& ex) {
    cout<<"Ha ocurrido un error desconocido:"<<ex.what()<<endl;
}

v2 = v3;
//Compruebo que algunas operaciones se realicen correctamente
v3[2] = true;
cout<<"v2="<<v2<<"", v3="<<v3<<endl;

//Operacion & (and) entre v1 y v2, solo permitida para vectores de igual tamaño
v2 = v1 & v2;
cout<<"v2="<<v2<<endl;
//Muestro los resultados de la operacion

return 0;
// Fin del programa
```

```
Error al intentar aplicar v1 & v2...Error en tamanios
v2=[0, 1, 0], v3=[0, 1, 1]
v2=[0, 1, 0]

Process returned 0 (0x0)   execution time : 0.095 s
Press any key to continue.
```