#include<iostream>

using namespace std;

//void dosomething() {

// throw "Heyy error";

//}

//

//

//void main() {

//

// try

// {

// dosomething();

// //

// //

// //

// }

// catch (const char\*ex)

// {

// cout << ex << endl;

// }

//

//

//}

//Stack Unwinding

//void foo(bool x) {

// char\* data = new char[1024];//address lost

// string text = "Hello World";

// if (x) {

// throw "Boom";

// }

// int x2 = 0;

// cout << "Deleted data" << endl;

// delete[]data;

//}

//

//

//void main() {

// try {

// foo(true);

// //

// //

// //

// }

// catch (const char\* ex) {

// cout << "error : " << ex << endl;

// }

//}

//int& last() {

// cout << "Start Last" << endl;

// cout << "Last throw exception" << endl;

// int x = 10;

// throw "Hey error in last";

// cout << "End Last" << endl;

// return x;

//}

//

//int& third() {

// cout << "Start Third";

// return last();

// cout << "End Third" << endl;

//}

//

//int& second() {

// cout << "Start Second" << endl;

// try {

// return third();

// }

// catch (double ex) {

// cout << "Second caught exception : " << ex << endl;

// }

// cout << "End Second" << endl;

//}

//

//int& first() {

// cout << "Start First" << endl;

// try {

// return second();

// }

// catch (const char\* ex) {

// cout << "First caught exception char\* : " << endl;

// }

// catch (double ex) {

// cout << "First caught exception double : " << endl;

// }

// cout << "End First" << endl;

//}

//

//

//void main() {

// cout << "Main Start" << endl;

// try {

// cout << first() << endl;

// }

// catch (const char\* ex) {

// cout << "Error : " << ex << endl;

// }

// cout << "Main End" << endl;

//}

class Exception {

protected:

string text;

string source;

int line;

string date;

public:

Exception(string text,string source,int line,string date)

: text(text),source(source),line(line),date(date)

{

}

void printMessage() const {

cout << "+++++++++++++ ERROR INFO +++++++++++++" << endl;

cout << "Error Content : " << text << endl;

cout << "Source : " << source << endl;

cout << "Line number : " << line << endl;

cout << "Date : " << date << endl;

}

};

class OutOfRangeException :public Exception {

public:

OutOfRangeException(string text,string source,int line,string date)

:Exception(text,source,line,date)

{

}

};

class InvalidArgumentException :public Exception {

public:

InvalidArgumentException(string text, string source, int line, string date)

:Exception(text, source, line, date)

{

}

};

template<typename T>

class Array {

private:

T\* data;

int size = 0;

public:

int getSize()const {

return size;

}

Array<T>& operator=(const Array<T>& other) = delete;

Array(const Array<T>&other) = delete;

Array(int size) throw(InvalidArgumentException)

{

if (size < 0 || size>1000000) {

throw InvalidArgumentException("Size should be between 0 and 1 billion", \_\_FILE\_\_, \_\_LINE\_\_, \_\_DATE\_\_);

}

this->size = size;

data = new T[size];

}

T& operator[](int index) throw(OutOfRangeException) {

if (index < 0 || index >= size) {

throw OutOfRangeException("Out of range exception", \_\_FILE\_\_, \_\_LINE\_\_, \_\_DATE\_\_);

}

}

~Array()

{

delete[]data;

}

};

void main() {

try {

Array<string>a(10);

cout << a[400] << endl;

}

catch (const Exception& ex) {

ex.printMessage();

}

}