#include <iostream>

using namespace std;

int\* doubleCapacity(const int\* list, int size) {

int \*ptr = new int[size];

for (int i = 0; i < size; i++) {

ptr[i] = list[i];

}

return ptr;

}

int main() {

int size, \*list, \*newlist, doubleSize, value;

cout << "Enter the size of your array: ";

cin >> size;

list = new int[size];

cout << "Enter" << size <<" integer values: ";

for (int i = 0; i < size; i++) {

cin >> value;

\*(list + i) = value;

}

doubleSize = size \* 2;

newlist = doubleCapacity(list, doubleSize);

cout << "The array has now doubled, enter more integers: ";

for (int i = size; i < doubleSize; i++) {

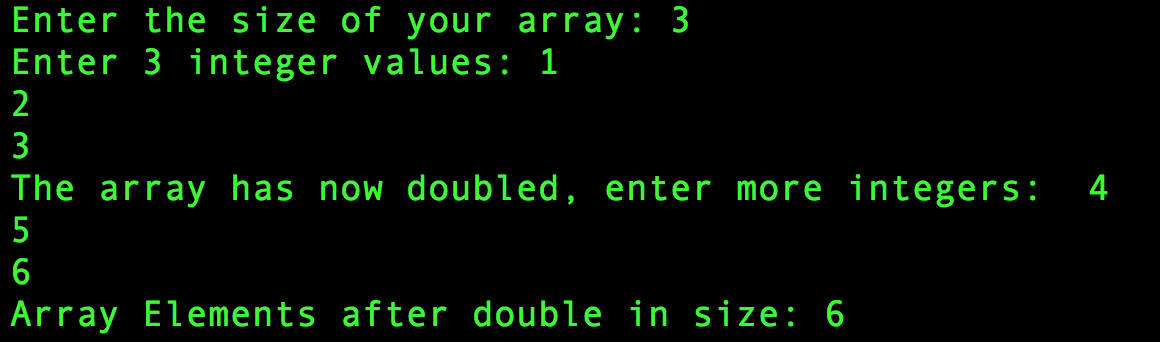
cin >> value;

\*(newlist + i) = value;

}

cout << "Array Elements after double in size: " << doubleSize << endl;

}



#include <iostream>

using namespace std;

class Rectangle2D {

private:

double x, y, width, height;

public:

Rectangle2D(){

x = 0;

y = 0;

width = 1;

height = 1;

}

Rectangle2D(double x1, double y1, double w, double h) {

x = x1;

y = y1;

width = w;

height = h;

}

// get set

void setX(double input) {

x = input;

}

void setY(double input) {

y = input;

}

void setWidth(double input) {

width = input;

}

void setHeight(double input) {

height = input;

}

double getX() {

return x;

}

double getY() {

return y;

}

double getWidth() {

return width;

}

double getHeight() {

return height;

}

double getArea() {

return width \* height;

}

double getPerimeter() {

return 2\*(width+height);

}

bool contains(double x, double y) {

double pX = x;

double pY = y;

if (pX < (x + (.5 \* width)) && (pX > (x - (.5 \* height))) && (pY < (y + (.5 \* height))) && (pY > (y - (.5 \* height)))) {

return true;

} else {

return false;

}

}

};

#include <iostream>

using namespace std;

template <typename T>

class Stack {

public:

bool empty() const {

return elements.empty();

}

T peek() const {

return elements.back();

}

void push(T value) {

elements.push\_back(value);

}

void pop() {

elements.pop\_back();

}

T getSize() const {

return elements.size();

}

T getData(int value) {

return elements.at(value);

}

private:

vector elements;

};