圆

时间限制:1s 空间限制:128MB

题目描述

结构体$Circle$表示一个圆，(x,y)是圆心的坐标，r是圆的半径。$CircleSet$类表示一个圆的集合，类中有一个数组circles[]来存储集合中所有圆，**集合中所有圆的半径互不相同。以下描述中，圆p指的是circles[p]。(0-base)**

你需要实现以下函数：

- $CircleSet$类中的析构函数。
- $CircleSet$类中的下标运算符重载$operator[]$：设set是$CircleSet$类的对象，$set[p]$返回圆p的半径。
- $CircleSet$类中的函数$bool\ checkContaining(int\ p, int\ q)$：如果圆$p$严格包含(不相切)圆$q$，返回1，否则返回0。
- $CircleSet$类中的函数$int\ getCircleContainingQ(int\ q)$：返回集合中严格包含圆$q$的半径最小的圆的半径。（保证一定存在）

  特别注意：

- **圆p严格包含圆q的充要条件：圆p的半径大于圆q的半径 且 两圆心的距离小于两圆半径之差的绝对值。**
- **如果你需要比较a和$\sqrt{b}$的大小，为了避免浮点误差，建议你比较a^2和b的大小。**
- **不能在类中定义新的成员变量，不能修改类中已经给定的函数。**
- **如果你没有实现某个函数，请把主函数中调用该函数的代码注释掉再提交，以保证代码可以编译。除此之外，不能修改主函数中的其他内容。**

```
#include <iostream>
#include <iomanip>
using namespace std;

class CircleSet {
  friend istream &operator>>(istream &in, CircleSet &obj);
  private:
    struct Circle {
      long long x, y, r;
      Circle(long long _x = 0, long long _y = 0, long long _r = 0): x(_x),
y(_y), r(_r) {}
    };
    int count;//集合中圆的数目
    Circle *circles;//集合中所有圆
```

```
  public:
    CircleSet(int n): count(n) {
      circles = new Circle[n];
    }
};

istream &operator>>(istream &in, CircleSet &obj) {
  for (int i = 0; i < obj.count; ++i) {
    in >> obj.circles[i].x >> obj.circles[i].y >> obj.circles[i].r;
  }
  return in;
}

int main() {
  int n, m, type;
  cin >> n >> m;
  CircleSet set(n);
  cin >> set; //输入集合中的所有圆
  while (m--) {
    int type, p, q;
    cin >> type;
    if (type == 1) {
      //do nothing
    } else if (type == 2) {
      for (int i = 0; i < n; ++i) {
        cout << set[i] << ' ';
      }
      cout << endl;
    } else if (type == 3) {
      cin >> p >> q;
      cout << set.checkContaining(p, q) << endl;
    } else if (type == 4) {
      cin >> q;
      cout << set.getCircleContainingQ(q) << endl;
    }
  }
  return 0;
}
```

数据范围

为了分别评测每个函数的实现，每个测试点只对应一个函数。一共有20个测试点来测试你的程序。

圆的横坐标、纵坐标和半径都是[-10000,10000]范围内的整数。

测试点编号	$type$	n	m	Hint	分值
$1-4$	1	100	1	测试析构函数	20%
$5-8$	2	100	1	测试下标运算符重载	20%
$9-14$	3	100	10	测试函数bool checkContaining(int p, int q)	30%
$15-20$	4	100	10	测试函数int getCircleContainingQ(int q)	30%

输入格式

第一行有两个整数n和m。

之后有m行，每行的第一个整数$type$表示测试类型。

$type=1$时，测试析构函数。

$type=2$时，测试下标运算符重载。

$type=3$时，$type$后面有两个整数p和q，测试函数bool checkContaining(int p, int q)。

$type=4$时，$type$后面有一个整数q，测试函数int getCircleContainingQ(int q)。

保证一个测试文件中的$type$都相等。

输出格式

具体输出格式请看主函数。

样例输入

```
10 10
0 0 1
0 0 9
0 0 2
0 0 6
0 0 8
0 0 7
9 0 12
-9 0 13
0 9 10
0 -9 11
1
2
3 0 1
3 1 0
3 5 6
3 9 0
3 9 2
4 0
```

```
4 2
4 5
```

样例输出

```
1 9 2 6 8 7 12 13 10 11
0
1
0
1
0
2
6
8
```