

题目描述：

篮球（5V5）比赛中，每个球员拥有一个战斗力，每个队伍的所有球员战斗力之和为该队伍的总体战斗力。现有 10 个球员准备分为两队进行训练赛，教练希望 2 个队伍的战斗力差值能够尽可能的小，以达到最佳训练效果。给出 10 个球员的战斗力，如果你是教练，你该如何分队，才能达到最佳训练效果？请输出该分队方案下的最小战斗力差值。

输入描述：

10 个篮球队员的战斗力（整数，范围[1,10000]），战斗力之间用空格分隔，如：10 9 8 7 6 5 4 3 2 1

不需要考虑异常输入的场景。

输出描述：

最小的战斗力差值，如：1

补充说明：

示例 1

输入：

10 9 8 7 6 5 4 3 2 1

输出：

1

说明：

1 2 5 9 10 分为一队，3 4 6 7 8 分为一队，两队战斗力之差最小，输出差值 1。备注：球员分队方案不唯一，但最小战斗力差值固定是 1

```
import java.util.Scanner;
```

```
// 注意类名必须为 Main，不要有任何 package xxx 信息
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
        Scanner in = new Scanner(System.in);
```

```
        // 注意 hasNext 和 hasNextLine 的区别
```

```
        while (in.hasNextInt()) { // 注意 while 处理多个 case
```

```
            int[] player = new int[10];
```

```
            int sum = 0;
```

```
            for(int i = 0; i < 10; i++) {
```

```

        player[i] = in.nextInt();
        sum += player[i];
    }

    int count = 5;
    int res = Integer.MAX_VALUE;
    res = BFS(player, 0, res, 0, sum, count);
    System.out.println(res);
}

}

public static int BFS(int[] player, int index, int res, int add, int sum, int count) {
    if(count == 0) {
        res = Math.min(res, Math.abs(sum - add*2));
        return res;
    }
    for(int i = index; i < 10; i++) {
        res = BFS(player, i + 1, res, add + player[i], sum, count - 1);
    }
    return res;
}
}

```