小美的一个兼职是美团的一名跑腿代购员，她有 n 个订单可以接，订单编号是 1~n，但是因为订单的时效性，他只能选择其中 m 个订单接取，精明的小美当然希望自己总的获利是最大的，已知，一份订单会提供以下信息，跑腿价格 v，商品重量 w kg，商品每重 1kg，代购费用要加 2 元，而一份订单可以赚到的钱是跑腿价格和重量加价之和。小美可是开兰博基尼送货的人，所以自然不会在意自己会累这种事情。请问小美应该选择哪些订单，使得自己获得的钱最多。

请你按照选择的订单编号的从小到大顺序，如果存在多种方案，输出订单编号字典序较小的方案。

```
 import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
import java.util.Arrays;
import java.util.Comparator;
import java.util.PriorityQueue;

public class Main {
   public static void main(String[] args) throws IOException {
      BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
      String[] strNM = br.readLine().trim().split(" ");
      int n = Integer.parseInt(strNM[0]);
      int m = Integer.parseInt(strNM[1]);
      Node[] goods = new Node[n];
      for(int i = 0; i < n; i++){
         String[] params = br.readLine().trim().split(" ");
         int v = Integer.parseInt(params[0]);
         int w = Integer.parseInt(params[1]);
         goods[i] = new Node(i + 1, v, w);
      }
      Arrays.sort(goods, new Comparator<Node>() {
         @Override
         public int compare(Node node1, Node node2) {
            if(node1.income > node2.income){
               return -1;
            }else if(node1.income < node2.income){
               return 1;
            }else{
               return node1.id - node2.id;
            }
         }
      });
      PriorityQueue<Integer> pq = new PriorityQueue<>();
      for(int i = 0; i < m; i++)
         pq.offer(goods[i].id);
      for(int i = 0; i < m; i++)
```

```
            System.out.print(pq.poll() + " ");
        }
    }

    class Node {
        public int income;
        public int id;
        public Node(int id, int v, int w) {
            income = v + 2*w;
            this.id = id;
        }
    }
```

小美是美团的前端工程师，为了防止系统被恶意攻击，小美必须要在用户输入用户名之前做一个合法性检查，一个合法的用户名必须满足以下几个要求：

1. 用户名的首字符必须是大写或者小写字母。
2. 用户名只能包含大小写字母，数字。
3. 用户名需要包含至少一个字母和一个数字。

如果用户名合法，请输出“Accept”，反之输出“Wrong”。

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        int n = Integer.parseInt(br.readLine().trim());
        for(int i = 0; i < n; i++){
            String username = br.readLine().trim();
            if(judge(username.toCharArray()))
                System.out.println("Accept");
            else
                System.out.println("Wrong");
        }
    }

    // 判断用户名合法性
    private static boolean judge(char[] username) {
        if(username == null || username.length == 0) return false;
        // 判断首字符是不是字母
        if(!isAlpha(username[0]))
            return false;
        // 判断是否只有字母或数字，以及是否既有字母又有数字
        int alphaNum = 0, digitNum = 0;
        for(int i = 0; i < username.length; i ++){
```

```
            if(isAlpha(username[i])){
                alphaNum ++;
            }else if(isDigit(username[i])){
                digitNum ++;
            }else
                return false;
        }
        return alphaNum > 0 && digitNum > 0;
    }

    private static boolean isAlpha(char c) {
        return (c >= 97 && c <= 122) || (c >= 65 && c <= 90);
    }

    private static boolean isDigit(char c) {
        return c >= 48 && c <= 57;
    }
}
```