#include <iostream>  
#include <vector>  
#include <map>  
#include <limits>  
  
  
int INF = std::numeric\_limits<int>::max();  
  
class Element {  
private:  
 int wage\_;  
 int profit\_;  
  
public:  
 Element(int wage, int profit) : wage\_(wage), profit\_(profit) {};  
  
// int get\_profit() { return profit\_; }  
 int get\_profit() const { return profit\_; }  
  
// int get\_wage() {return wage\_; }  
 int get\_wage() const { return wage\_; }  
  
 int calculate\_cost() const { return wage\_ \* profit\_; }  
  
};  
  
// rekursja  
int DP\_FPLP(std::vector<Element> &elements, int limit, int stage) {  
 int cost = 0;  
 if (stage == 0)  
 return 0;  
 else  
 cost = DP\_FPLP(elements, limit, stage-1);  
  
 std::cout << "BREAK" << std::endl;  
}  
  
// orgnial  
void DP\_FPLP(std::vector<Element> &elements, int limit) {  
 DP\_FPLP(elements, limit, (int)elements.size());  
  
 std::cout << "BREAK" << std::endl;  
}  
  
  
int main() {  
 std::cout << "Hello, World!" << std::endl;  
 return 0;  
}