```cpp
#include <iostream>
#include <vector>
#include <memory>


class BadInput: public std::exception {};

bool isInRangeStart(const int num,const int vectorSize){
    return num < vectorSize && num >= 0;
}
bool isInRangeStop(const int num,const int vectorSize) {
    return num <= vectorSize && num >= 0;
}
template <class T>
std::vector<T> slice(std::vector<T> vec, int start, int step, int stop) {
    if(!isInRangeStart(start, vec.size()) || !isInRangeStop(stop,vec.size())){
        throw BadInput();
    }
    if (step <= 0){
        throw BadInput();
    }
    std::vector<T> ans;
    for(int i = start; i<stop;i+=step){
        ans.push_back(vec[i]);
    }
    return ans;
}
/**part A**/
```

```cpp
/**part B**/
class A {
public:
    std::vector<std::shared_ptr<int>> values;
    void add(int x) {
        std::shared_ptr<int> shared_ptr_to_x(new int(x));
        values.push_back(shared_ptr_to_x);
    }
    ~A() =default;
};
```