```cpp
#include <iostream>
#include <string>
#include <cstdio>
#include <list>
#include <sstream>

struct aaa
{

};

struct op : public aaa
{
        char operace;

        op(char a) : operace(a) { }
};

struct integer : public aaa
{
        int value;

        integer(int a) : value(a) { }
};

int main()
{
        std::ios::sync_with_stdio(false);

        std::string line;
        int R, C;
        while (scanf("%d %d", &R, &C) == 2)
        {
                std::list<int> list;
                if (R == 1)
                {
                        std::getline(std::cin, line);

                        std::cout << line << std::endl;


                        std::stringstream lineStream(line);
                        int a;
                        while (lineStream >> a)
                        {
                                list.push_back(a << 2);

                                char operation;
```

```cpp
				if (lineStream >> operation)
				{
					if (operation == '*')
						list.push_back(1);
					else if (operation == '+')
						list.push_back(2);
					else if (operation == '-')
						list.push_back(3);
				}
				else
					break;
			}

			std::list<int>::iterator i;
			for (i = list.begin(); i != list.end(); i++)
			{
				if (((*i) & 0x3) == '*')
				{
					std::list<int>::iterator intPred = i--;
					std::list<int>::iterator intPo = i++;

					int a = (*intPred) >> 2;
					int b = (*intPo) >> 2;

					i = list.insert(intPred, (a * b) << 2);
					list.erase(intPred, intPo++);
				}
			}

			/*for (i = list.begin(); i != list.end(); i++)
			{
				if (((*i) & 0x3))
				{
					int op = (*i) & 0x3;

					std::list<int>::iterator intPred = i--;
					std::list<int>::iterator intPo = i++;

					int a = (*intPred) >> 2;
					int b = (*intPo) >> 2;

					if (op == 2)
						i = list.insert(intPred, a + b);
					else
						i = list.insert(intPred, a - b);
					list.erase(intPred, intPo++);
				}
			}*/
```

```

                        std::cout << list.front() << std::endl;
                }
        }

        return 0;
}
```