

```
1: #include <iostream>
2: #include <string>
3: #include <cstdio>
4: #include <list>
5: #include <sstream>
6:
7: struct aaa
8: {
9:
10: };
11:
12: struct op : public aaa
13: {
14:     char operace;
15:
16:     op(char a) : operace(a) { }
17: };
18:
19: struct integer : public aaa
20: {
21:     int value;
22:
23:     integer(int a) : value(a) { }
24: };
25:
26: int main()
27: {
28:     std::ios::sync_with_stdio(false);
29:
30:     std::string line;
31:     int R, C;
32:     while (scanf("%d %d", &R, &C) == 2)
33:     {
34:         std::list<int> list;
35:         if (R == 1)
36:         {
37:             std::getline(std::cin, line);
38:
39:             std::cout << line << std::endl;
40:
41:
42:             std::stringstream lineStream(line);
43:             int a;
44:             while (lineStream >> a)
45:             {
46:                 list.push_back(a << 2);
47:
48:                 char operation;
```

```
49:         if (lineStream >> operation)
50:         {
51:             if (operation == "**")
52:                 list.push_back(1);
53:             else if (operation == "+")
54:                 list.push_back(2);
55:             else if (operation == "-")
56:                 list.push_back(3);
57:         }
58:         else
59:             break;
60:     }
61:
62:     std::list<int>::iterator i;
63:     for (i = list.begin(); i != list.end(); i++)
64:     {
65:         if (((*i) & 0x3) == "**")
66:         {
67:             std::list<int>::iterator intPred = i--;
68:             std::list<int>::iterator intPo = i++;
69:
70:             int a = (*intPred) >> 2;
71:             int b = (*intPo) >> 2;
72:
73:             i = list.insert(intPred, (a * b) << 2);
74:             list.erase(intPred, intPo++);
75:         }
76:     }
77:
78:     /*for (i = list.begin(); i != list.end(); i++)
79:     {
80:         if (((*i) & 0x3))
81:         {
82:             int op = (*i) & 0x3;
83:
84:             std::list<int>::iterator intPred = i--;
85:             std::list<int>::iterator intPo = i++;
86:
87:             int a = (*intPred) >> 2;
88:             int b = (*intPo) >> 2;
89:
90:             if (op == 2)
91:                 i = list.insert(intPred, a + b);
92:             else
93:                 i = list.insert(intPred, a - b);
94:             list.erase(intPred, intPo++);
95:         }
96:     }*/
```

```
97:
98:         std::cout << list.front() << std::endl;
99:     }
100: }
101:
102:     return 0;
103: }
```