

1) Escrever um programa q1.cpp que lê uma quantidade indeterminada de valores double, inserindo-os em um container, sem aceitar duplicados (isto é, se o valor já foi inserido anteriormente, não inserir de novo). Encerrar a entrada quando o valor digitado for 0.0 (zero), o qual não deve ser inserido. Após finalizar a entrada, o programa deve, exibir o tamanho e o maior valor contido, exibir os dados em ordem normal e ordem reversa.

Entrada	Saída
10.1 3.1 3.5 14.3 1.0 10.6 9.8 1.0 19.6 15.1 12.6 18.8 3.5 0.0	Tamanho:11 Maior valor: 19.6 Valores: 10.1 3.1 3.5 14.3 1.0 10.6 9.8 19.6 15.1 12.6 18.8 Reverso: 18.8 12.6 15.1 19.6 9.8 10.6 1.0 14.3 3.5 3.1 10.1

2) Implemente um emulador de uma **máquina baseada em pilha**, que executa operações inteiras simples. Todos os comandos são fornecidos em formato **explícito**, evitando qualquer necessidade de interpretar caracteres individuais. A máquina mantém uma **pilha de inteiros** e suporta os seguintes comandos:

- **PUSH X** - Empilha o inteiro X no topo da pilha.
- **ADD** - Desempilha os dois valores do topo, soma e empilha o resultado.
- **SUB** - Desempilha dois valores, realiza (segundo do topo – topo) e empilha o resultado.
- **MUL** - Desempilha dois valores, multiplica e empilha o resultado.
- **PRINT** - Desempilha e imprime o valor removido.

**Entrada:** A entrada começa com um número inteiro N ( $2 \leq N \leq 100$ ), indicando a quantidade de comandos. Nas próximas N linhas haverá sempre um comando válido, em um dos formatos:

- **PUSH X** — X é um inteiro (positivo, negativo ou zero)

### Saída

Para cada execução do comando **PRINT**, imprima o valor removido da pilha em uma linha separada.

Entrada	Saída
6 PUSH 4 PUSH 2 ADD PUSH 10 MUL PRINT	60

Entrada	Saída
10 PUSH 3 PUSH 8 PUSH 2 MUL ADD PUSH 5 SUB PUSH -4 MUL PRINT	-56

3) Implementar na biblioteca "lista.cpp" a seguinte função:

```
void insere_apos(const std::string &s1, const std::string &s2)
```

Ela recebe duas strings (s1 e s2) e deve inserir na lista o valor passado em s1 após o primeiro nodo que contêm o valor s2, se este existir. Caso o valor passado em s2 não esteja contido na lista, a função deve inserir s1 no final da lista.

**Exemplo:** l.insere\_apos("x", "y");

Neste exemplo, o valor "x" deve ser inserido na lista após "y", se existir "y" nesta lista. Se não existir nenhum "y", inserir no final da lista. Se existir mais de um "y", inserir depois da primeira ocorrência. Utilize o arquivo teste2.cpp para validar sua implementação