

Listas ligadas II

Nesta atividade você deverá fazer um programa que seja capaz de fazer soma e multiplicação de inteiros arbitrariamente grandes. Seu programa deverá ler, da entrada padrão, uma *conta* por linha, até que o fim do arquivo seja atingido. Cada *conta* compreende um número inteiro não-negativo com um número arbitrário de dígitos decimais, um operador de mais ou de vezes, e outro número inteiro não-negativo com um número arbitrário de dígitos decimais. Seu programa deve armazenar cada número inteiro numa lista ligada, um dígito decimal por nó. As operações de soma e multiplicação deverão ser implementadas manualmente por você, para operar com essas listas de dígitos e obter o resultado desejado de cada conta. Seu programa deve imprimir o resultado de cada conta em uma linha diferente da saída padrão. Cada número da entrada poderá ter até 32768 dígitos.

Este é um exercício de lista ligada! Portanto, se você está se perguntando se pode usar um vetor para armazenar os dígitos de um número em vez de uma lista ligada, você já sabe qual é a resposta para essa pergunta. É proibido usar qualquer vetor neste exercício programa! E não vale trapacear (por exemplo, substituir `x[i]` por `*(x + i)` para mascarar o uso do vetor ou fazer coisa parecida)!

Exemplos

Exemplo de entrada:

```
356719340560150932+71923471923740129
193284719283740928374091827340981234129837490*92381324786
127348968*1892374890121298374
24735698273458*0
```

Saída para a entrada do exemplo:

```
428642812483891061
17855898428322107993208140009375422158079625257089027140
240991989326060742748978032
0
```

Scripts

Para gerar a entrada de teste, você pode rodar

```
chmod 755 gen.py
./gen.py > entrada.txt
```

onde `gen.py` é o script em Python3 a seguir.

```
#!/usr/bin/python3

import random as rd

rd.seed(2017)

for i in range(100):
    s = rd.randint(1, rd.randint(10, rd.randint(100, 32768)))
    for j in range(s):
        print(rd.randint(0, 9), end='')
        if (rd.randint(0, 1) == 0):
            print('+', end='')
        else:
            print('*', end='')
    s = rd.randint(1, s)
    for j in range(s):
        print(rd.randint(0, 9), end='')
    print()
```

Para gerar a saída-gabarito correspondente à entrada gerada pelo script acima, você pode rodar (numa linha só) os comandos

```
cat entrada.txt | while read in; do
    echo "$in" | BC_LINE_LENGTH=0 bc;
done > gabarito.txt
```

Para gerar a saída do seu programa correspondente à entrada gerada pelo script `gen.py`, você pode rodar

```
seu_programa < entrada.txt > saida.txt
```

Finalmente, para verificar se seu programa funcionou corretamente, faça

```
diff saida.txt gabarito.txt
```

Se o último comando não gerar saída nenhuma quer dizer que os arquivos `saida.txt` e `gabarito.txt` são idênticos, e que seu programa está correto.