Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação

SCC-202 - Algoritmos e Estruturas de Dados I

Responsável: Prof. Gustavo Batista
gbatista@icmc.usp.br

Estagiário PAE: Vinícius Souza
vsouza@icmc.usp.br

Monitor PEEG: Luís Fernando Dorelli de Abreu
lfdorelli@gmail.com

Projeto 4 – Aritmética de Grandes Números

Os critérios de correção deste exercício são:

80% - Implementação do TAD e da aplicação, dos quais
- 25% - Implementação da soma
- 25% - Implementação da subtração
- 25% - Implementação da multiplicação
- 15% - Implementação da potencia
- 10% - Leitura e formatação da resposta

10% - Modularização do código
10% - Documentação do código

O projeto deve ser feito individualmente. Não é obrigatório o uso dos TADs desenvolvidos em aula, e será aceita a utilização de implementações padrão de listas (mais detalhes adiante). Quaisquer programas similares terão nota zero independente de qual for o original e qual for a cópia. A data de entrega é 05/11/2012. Trabalhos atrasados terão a nota descontada em 2 pontos por dia de atraso. Deve ser utilizada a linguagem de programação C/**C++**.

Submissão

Os trabalhos deverão ser enviados a um sistema de submissão automática. Para a submissão automática, todo o código deve estar em um único arquivo. A submissão deverá ser feita utilizando o sistema Boca (http://sites.labic.icmc.usp.br/boca/scc202/). Haverá uma entrada para cada caso de teste. Você deverá enviar seu programa para cada

um deles. O usuário de cada aluno é a concatenação se seus primeiros nomes (de e van não contam, utilize o nome seguinte). Por exemplo, Alfredo de Onofre terá o usuário alfredoonofre. A senha é o número USP. Apesar dos casos de teste estarem separados entre as operações, é recomendável enviar o programa que trata todas as operações em todos os casos (não faça programas diferentes para casos diferentes).

Além disso, os projetos (código fonte e makefile) devem ser compactados e enviados pelo site do TIDIA (escaninho), no link http://tidia-ae.usp.br/portal. Para a entrega no TIDIA, organize seu projeto em módulos da maneira usual.

Utilização de Listas Ligadas

A utilização de listas ligadas para implementar a estrutura de dados pedida no trabalho é **obrigatória**. Entretanto, será permitida a utilização da STL (*Standard Template Library*), que contém essa estrutura já implementada, da linguagem C++. Isso não significa que o código deva ser desenvolvido com orientação a objetos – é apenas um recurso para simplificar o projeto. Um pequeno tutorial sobre a utilização da estrutura *list* pode ser encontrado no repositório do TIDIA, assim como um exemplo funcional. A utilização da STL e da linguagem C++ não é obrigatória.

Enunciado

Neste trabalho você deverá implementar uma estrutura de dados capaz de armazenar, operar e imprimir grandes números, em qualquer base. Na base 10, um número é representado por uma sequência de dígitos, com valores entre 0 e 9, e, possivelmente, um sinal. Em uma base *B* qualquer, um número é representado por uma sequência de dígitos com valores entre 0 e *B*-1, e o sinal.

Você deve implementar um programa capaz de ler diversas linhas, cada uma contendo uma operação. Cada operação será passada da seguinte forma: *argumento1_op_argumento2_base_separador*, onde

- **argumento1**: *string* representando um número na base 10, com ou sem sinal. Até 1000 dígitos.
- **Op:** caractere, pode ser +, -, * ou ^ (soma, subtração, multiplicação e potencia, respectivamente);
- **argumento2**: *string* representando um número na base 10, com ou sem sinal. Até 1000 dígitos;
- **base**: inteiro que representa base na qual o resultado final deve ser impresso. ($2 \leq$ base $\leq 10^4$);
- **separador**: *string* que deve ser impressa entre cada dígito do número se a base for maior do que 10. Se a base for menor do que 10, o separador deve ser ignorado;

- Há necessariamente um espaço entre argumento1 e op, e entre op e argumento2, entre argumento2 e base e entre base e separador. Não há espaços entre os sinais e os números

É opção do aluno a escolha entre uma implementação que apenas realize as contas em uma base específica e depois realize a conversão para outras bases ou uma implementação que trabalhe diretamente com qualquer base.
Para cada linha lida, o programa deverá imprimir o resultado da operação. Para mais detalhes da entrada/saída, veja os casos de exemplo. A entrada deve ser lida até o fim de arquivo (EOF). **Não será necessário realizar o tratamento de erros, mas deve-se adicionar um breve comentário sobre cada função onde poderiam ocorrer problemas comentando-os.**

Exemplo de Entrada

```
450 - 75 2 !!
78 - 79 16 ***
999999999 ^ 3 22 &
-1294184129841294821491248l2 + 1294184129841294821491248l2 1000 *
12844892948 * -28173128739821739813 10 !!!
2727 ^ 77 1000 :
-241412412424155555 + 3324143221 10 **
-8 ^ 2000 10 !
```

Exemplo de Saída

```
101110111
-1
1&9&4&2&7&15&12&0&18&15&6&12&13&20&11&11&2&8&1&4&3
0
-36188082267323239250109453872 4
3:529:844:887:219:816:335:534:856:909:347:835:102:43:790:724:24:241:596:4
28:249:220:355:386:495:748:106:497:460:770:90:15:124:998:819:364:705:73:8
75:971:289:191:838:471:585:968:133:352:435:723:44:110:462:789:190:897:262
:452:955:601:473:691:514:548:636:464:674:277:119:484:686:476:350:196:647:
893:603:941:506:769:926:413:269:913:912:317:92:163:847
-241412409100012334
151347058230423707251341006732939195542348235662207750883638941664688930 6
993564534635830817676552455824162236150182627025523267446014684388515185 4
526108723851319250149779444829108931948648700394505490672981707219397118 2
719527789915234880110767164459088215765989790571534230657466816965835469 9
728703352747795407934837779271083755217530954282733292552820320384388452 7
366058498540970098661991758472895321264396779463236772187411951760311436 0
552024668199393907504691184161741062722198726771372433264644606105316057 2
807286503464245558500643221584631072641658731573097806459864337910226647 8
295694942840552297515997366197537288481880906923187735747021746988432990 8
683137306257570375794214926339926478753004889715437381938113609710511860 7
145954825590031647682047062652157439770794084025717418220995063370720546 6
```

651685466337470969350111480651084319752896540406273448746070098073153637711047169858694818610407398588834446676357099025935816107553760899412918841208218112272101571562981484028192147319292708955088212300964519681942673642842293257685034895099919352152310816490238943468681331108033283105412689503210430615429388761900128607792101045802764036928870342460867543142795664589429370819993239282948771664062223104536778920685573405405974590947621725444924404788767939600029240596345162993942014559414826151644985729036671441888555988467631460900650790052677466442375820863285851010603592000859395609822211391904102903207364712236794311036044733748555197204542785337080188658241477946611660336334854296138799639931239314000147310494322396767397430306893744422672660978572654020447591376885001622370857778426050489989977006563465452645156300424843821640659966433598162056145286906391871672784803408203476443527453979020321330042658744033925567453617804988175836550410481014347167049425945670231608006490728693 76