Instituto de Computação - Unicamp

MC102 - Algoritmos e Programação de Computadores

Laboratório 09 - Operações com Números Grandes

Prazo de entrega: 21/04/2017 23:59:59

Peso: 1

Professor: Eduardo C. Xavier Professor: Guido Araújo

Monitor: Arthur Pratti Dadalto Monitor: Cristina Cavalcante Monitor: Klairton de Lima Brito Monitor: Luís Felipe Mattos

Monitor: Paulo Finardi

Monitor: Paulo Lucas Rodrigues Lacerda

Monitor: Pedro Alves

Monitor: Renan Vilas Novas

Descrição

Em um computador, as variáveis que armazenam números possuem um tamanho limitado, ocupando em geral 32 bits ou 64 bits. Isto restringe os valores máximos e mínimos que podem ser representados em uma variável.

Na maioria dos computadores, os tipos inteiros " int " e " long int " servem para criar variáveis com tamanho de 32 e 64 bits respectivamente.

Com estes tamanhos, os maiores números inteiros positivos que podem ser representados são:

- 2^31 1 = 2147483647 para variáveis do tipo " int "
- 2^63 1 = 9223372036854775807 para variáveis do tipo "long int "

Isto é uma restrição séria para certas aplicações que precisam lidar com números maiores do que estes. Para uma variável do tipo " int " por exemplo, se somarmos (1 + 2147483647) e atribuirmos este valor para ela, teremos como resultado o valor -2147483648, ao invés de 2147483648 pois este último valor não pode ser representado na variável. Este erro é conhecido como "overflow" pois o resultado da operação não pode ser representado considerando o tamanho da variável.

Uma técnica usada para representar números maiores que estes, é utilizar um vetor para representar os números, onde cada dígito de um número ocupa uma posição do vetor. Por exemplo, podemos criar um vetor

para representar o valor (9223372036854775808) que não poderia ser representado em uma variável do tipo "long int".

A inconveniência disso é que temos que implementar as operações aritméticas que desejamos utilizar. Outro inconveniente é que o problema de "overflow" ainda existe, apesar de que podemos agora representar números maiores. Para o exemplo do vetor "v" visto acima, a operação

causaria "overflow" da mesma forma, pois os números que podemos representar estão restritos a 30 dígitos.

Neste laboratório você deve implementar um programa que realiza as operações soma e subtração sobre números inteiros positivos grandes,

que devem ser representados por um vetor de tamanho 30.

O programa deve primeiramente ler duas sequências de dígitos, representando os números, intercalados por um símbolo de + ou - representando a operação de soma ou subração, respectivamente, e então realizar a soma ou subtração dos números e imprimir o resultado.

O programa também deve levar em consideração o problema de "overflow" descrito em relação ao resultado da operação, e imprimir um erro caso ele ocorra.

Objetivo

O seu objetivo é fazer um programa que calcula as operações de soma e subtração entre dois inteiros positivos grandes de até 30 dígitos.

O seu programa deve possuir um controle de erro para verificar se o valor do resultado tem mais de 30 dígitos, sem contar o possível símbolo ' - '.

Entrada

A entrada consiste de duas sequências de dígitos de no máximo 30 caracteres representando dois números inteiros positivos, e um símbolo de + ou - entre os dois números, representando a operação desejada.

Você pode usar a seguinte sequência para ler uma cadeia de caracteres até a quebra de linha.

```
char c;
do{
   scanf("%c", &c);
}while(c!='\n');
```

Saída

A saída é composta de três números. Você deve imprimir os dois números

da entrada com 30 dígitos. Caso o número não tenha 30 dígitos, você deve completar com zeros à esquerda até que o número tenha. O número final deve ser o resultado da operação dos números da entrada. O resultado deve conter o símbolo - no começo, caso seja negativo.

Caso o resultado seja um número com mais de 30 dígitos, indicando que ocorreu um "overflow" no resultado, a saída deverá ser a mensagem "overflow".

Após a impressão da saída deve-se imprimir o caractere '\n'.

Exemplos

Teste 01

Entrada

13

+

8

Saída

Teste 02

Entrada

13

-

8

Saída

Teste 03

Entrada

8

_

13

Saída

Teste 04

Entrada

81

-

132

Saída

Teste 05

Entrada

9876543210 + 123456789

Saída

0000000000000000000009876543210 00000000000000000000000123456789 9999999999

Teste 06

Entrada

987654321098765432109876543210 + 987654321098765432109876543210

Saída

987654321098765432109876543210 987654321098765432109876543210 overflow

Para mais exemplos, consulte os testes abertos no Susy.

Observações

- O número máximo de submissões é 10;
- O seu programa deve estar completamente contido em um único arquivo denominado lab09.c;
- Para a realização dos testes do SuSy, a compilação dos programas desenvolvidos considerar o comando:__ gcc -std=c99 -pedantic -Wall -o lab09 lab09.c;
- Você deve incluir, no início do seu programa, uma breve descrição dos objetivos do programa, da entrada e da saída, além do seu nome e do seu RA;
- Indente corretamente o seu código e inclua comentários no decorrer do seu programa.

Critérios importantes

Independentemente dos resultados dos testes do SuSy, o não cumprimento dos critérios abaixo implicará em nota zero nesta tarefa de laboratório.

• O único header aceito para essa tarefa é o stdio.h.