Universidade Federal de Goiás
Curso de Sistemas de Informação
Introdução à Programação- 2022-2
Prova P4

Prof. Thierson Couto Rosa

Sumário

1 Melhores Cientes — 5,0 pontos

Uma dada loja mantém os seguintes dados sobre seus clientes:

- nome do cliente — no máximo 300 caracteres;
- CPF do cliente (cpf) — 11 caracteres;
- número de vezes em que o cliente comprou produtos vendidos pela empresa (*vezes*);
- valor médio dos produtos comprados pelo cliente (*val_med*);
- atraso médio em dias dos pagamentos do cliente para a empresa (atr).

A loja pretende gerar uma lista ordenada dos seus melhores clientes, onde o melhor cliente deve ocorrer em primeiro lugar na lista, o segundo melhor cliente em segundo lugar, e assim por diante. Dados dois clientes a e b quaisquer, a empresa define a seguinte ordem de precedência (ou ordem de ser "melhor que") de a em relação a b, denotada por $a \prec b$ (a precede b):

$$a \prec b \text{ se} \begin{cases} 2(a.vezes \times a.val\_med) - a.atr > 2(b.vezes \times b.val\_med) - b.atr & ou \\ 2(a.vezes \times a.val\_med) - a.atr = 2(b.vezes \times b.val\_med) - b.atr \;\&\&\; a.cpf < b.cpf \end{cases} \tag{1}$$

Onde $a.cpf < b.cpf$ significa que a cadeia de caracteres que forma o cpf de a é lexicograficamente[1] menor que a cadeia de caracteres que forma o cpf de b.

A empresa pede que você escreva um programa para ler os dados de vários clientes e para gerar uma lista dos clientes segundo a ordem de precedência por ela definida.

Entrada

A primeira linha da entrada contém um número inteiro indicando o número de clientes da empresa. O dados de cada cliente estão dispostos em cinco linhas. A primeira linha contém o nome do cliente que pode ter no máximo 300 caracteres. A segunda linha contém o CPF do cliente, uma string, com 11 caracteres numéricos. A terceira linha contém o número de vezes que o cliente fez compras com a empresa. A quarta linha contém o valor médio das compras que o cliente fez, considerando as vezes que ele comprou da empresa. A quinta linha corresponde ao atraso médio, em dias dos pagamentos que o cliente fez nas vezes em que comprou da empresa.

Saída

A saída corresponde ao ranque dos clientes da empresa, ditado pela ordem de precedência acima. Os dados de cada cliente devem aparecer como na entrada, um por linha, e obedecendo a ordem da entrada, isto é, nome seguido do CPF, que por sua vez é seguido por número de vezes que o cliente comprou, em seguida pelo valor médio das compras e finalmente pelo atraso médio. Entre os dados de um cliente e de outro e também após o último cliente impresso deve haver uma linha em branco.

[1] Lembre-se que comparação de strings não pode ser feita com os operadores relacionais, ==, > ou <. É necessário usar uma função apropriada para isso, declarada em string.h, ou escrever sua própria função que compara caracter por caracter das duas strings que estão sendo comparadas.

Exemplos

Entrada
4 Joao Alves da Silva 40137811104 20 30.5 0 Maria Lucia Pereira 40971123409 10 308.78 5 Pedro Vieira Andrade 00572389020 20 30.5 4 Luciana Souza 00953472101 10 61.0 4
Saída
Maria Lucia Pereira 40971123409 10 308.78 5 Joao Alves da Silva 40137811104 20 30.50 0 Pedro Vieira Andrade 00572389020 20 30.50 4 Luciana Souza 00953472101 10 61.00 4

2 Melhores Clientes — Versão 2 — 2,0 pontos

Com o programa que você fez para a Questão 1, a loja pôde dar incentivos promocionais a seus seus melhores clientes. Isso fez com com que mais pessoas se tornassem clientes da empresa. Isso é ótimo! Mas como o número de clientes passou a aumentar com frequência, ficou complicado para a loja ter que contar o número de clientes para informar esse número como primeiro dado de entrada. A empresa quer agora que você mude o programa para que a entrada não seja mais controlada pelo número de clientes, mas sim por fim de arquivo. É obvio que não basta mudar o comando de repetição para processar a entrada. Essa nova exigência faz com que você não saiba mais qual o tamanho do vetor a ser usado para armazenar os dados dos clientes para ordená-los depois. Você terá que alocar dinamicamente a memória de modo a permitir que o vetor que armazena o ranque possa crescer de tamanho durante a leitura. Você deve realocar o vetor sempre que o numero de clientes lidos exceder o tamanho atual do vetor.

Entrada

Semelhante à entrada da especificada Questão 1, mas sem a primeira linha que indica o número de clientes.

Saída

A mesma da Questão 1.

Exemplos

Entrada
4 Joao Alves da Silva 40137811104 20 30.5 0 Maria Lucia Pereira 40971123409 10 308.78 5 Pedro Vieira Andrade 00572389020 20 30.5 4 Luciana Souza 00953472101 10 61.0 4
Saída
Maria Lucia Pereira 40971123409 10 308.78 5 Joao Alves da Silva 40137811104 20 30.50 0 Pedro Vieira Andrade 00572389020 20 30.50 4 Luciana Souza 00953472101 10 61.00 4

3 New Strcat - 3,0 pontos

A função `char* strcat(char* s1, char* s2)` é utilizada para emendar uma string apontada por `s1` em com outra string apontada por `s2`. O caracter '\0' da primeira string é sobreposto com o primeiro caractere da segunda string e os demais caracteres da string apontada por `s2` são inseridos em sequência ao final da string apontada por `s1`. por fim, o caractere '\0' é inserido ao final da string apontada por `s1`. Por exemplo, suponha duas strings, recebidas pela função `strcat()` como ilustradas abaixo:

`s1` → Ola, tudo '\0'
`s2` → bem com voce?'\0'

Após a chamada a `strcat`, o resultado é o seguinte:

`s1` → Ola, tudo bem com voce?'\0'
`s2` → bem com voce?'\0'

Ou seja a string apontada por `s2` continua como era no início da chamada e a string apontada por `s1` passa a ser a primeira string emendada com a segunda. Esse comportamento da função `strcat` tem dois inconvenientes:

- é necessário que haja espaço suficiente ao final da string apontada por `s1` para caber todos os caracteres da string apontada por `s2` e mais o caractere de fim de string;
- a função modifica a string original apontada por `s1`, o que nem sempre é o desejável.

A função `strcat()` retorna o endereço da primeira string, isso é o mesmo endereço armazenado em `s1`.

Pede-se que você escreva uma nova função cujo cabeçalho (*header*) deve ser o seguinte, `char * newstrcat(char* s1, char* s2)`. Essa função deve manter as duas strings recebidas como parâmetros intactas. Deve tentar reservar espaço na memória para armazenar a emenda da string apontada por `s1` com a string apontada por `s2`, realizar a emenda nessa nova área de memória e retornar o endereço de início dessa nova área. Caso a função não consiga reservar memória para a emenda, ela deve retornar zero (NULL).

Veja que a função `newstrcat()` definida acima não apresenta mais os inconvenientes listados acima, porém ela tem um inconveniente de que o programador quando não precisar mais da string emendada deve liberar essa área.

Escreva um programa para ler vários casos de testes. Cada caso de teste é formado por duas linhas, cada uma contendo uma string. Para cada caso de teste seu programa deve chamar a função `newstrcat()` passando como primeiro parâmetro a string da primeira linha e como segundo parâmetro, a string da segunda linha. Seu programa deve testar se a função `newstrcat()` conseguiu realizar seu trabalho, e nesse caso, deve imprimir a string emendada e depois liberar a área de memória por ela ocupada. Caso a função `newstrcat()` não tenha conseguido emendar as duas strings, seu programa deve emitir a mensagem NAO HOUVE MEMORIA PARA A CONCATENACAO e passar ao próximo caso de teste. A área de memória alocada pela função `newstrcat()` deve ser de tamanho apenas suficiente para armazenar a concatenação (emenda) das duas strings e o caractere de término de string.

Entrada

A primeira linha da entrada contém o número de casos de teste. Em seguida há duas linhas para cada caso de teste, cada um a contendo uma string que termina com o caractere de quebra de linha. O tamanho de cada string na entrada é de no máximo 200 caracteres.

Saída

Para cada caso de teste seu programa deve imprimir uma linha com a concatenação da primeira string com a segunda, nesta ordem, ou a frase NAO HOUVE MEMORIA PARA A CONCATENACAO, caso não seja possível a concatenação.

Exemplo

Entrada
3 Ola, tudo bem com voce? Universidade Federal de Goias - INF - Sistemas de Informacao Junção de strings em C
Saída
Ola, tudo bem com voce? Universidade Federal de Goias - INF - Sistemas de Informacao Junção de strings em C