

Universidade Federal de Goiás
Curso Sistemas de Informação
Introdução à Programação- 2020-1
Prova P3

Prof. Thierson Couto Rosa

A questão quatro é opcional e sua pontuação será utilizada para ser somada na nota da prova que tiver menor valor no fim do semestre.

Sumário

1	Inversão de Letras - 3,0 pontos	2
2	Quantidade de Elementos Distintos - 3,0 pontos	3
3	Positivos e negativos - 4,0 pontos	4
4	Strcmp() Nerd - Questão opcional - 3,0 pontos	5

1 Inversão de Letras - 3,0 pontos

Uma professora alfabetizou seus alunos, mas quer treiná-los mais com a manipulação de letras nas palavras. Ela pensou em uma brincadeira que geralmente as crianças gostam: inverter as palavras. Ela pretende fornecer para as crianças uma lista de frases com palavras escritas de forma invertida. Cada criança deve reescrever cada frase com as letras das palavras na ordem correta. Ela quer que você escreva um programa que leia frases escritas em ordem invertida e que as escreva na ordem correta. Ela pretende usar seu programa para gerar um gabarito de correção da lista de exercícios que ela quer criar.

Entrada

A entrada é formada por zero ou mais linhas. Uma linha da entrada contém uma frase escrita com as letras invertidas. A entrada termina por fim de arquivo. Você pode assumir que toda string lida terá no máximo 499 caracteres.

Saída

Para cada linha lida na entrada, deve ser escrita a frase correspondente com as letras na ordem correta.

0,5 Ponto Extra

Se você implementar a inversão da frase através de uma função do tipo void que recebe como parâmetro um ponteiro para char e o programa estiver funcionando corretamente, você ganhará mais 0,5 ponto nessa questão.

Exemplo

Entrada
sahnilab e sotilurip maroda sacnairc saditreveni oas saditrevid sartel moc sesarf
Saída
criancas adoram pirulitos e balinhas frases com letras divertidas sao invertidas

2 Quantidade de Elementos Distintos - 3,0 pontos

Escreva um programa que leia vários vetores e para cada vetor imprima o número de elementos que ocorrem apenas uma vez no vetor.

Entrada

A primeira linha da entrada contém o número C de casos de teste. Para cada caso de teste há uma linha contendo um inteiro $N \leq 500$ que corresponde ao tamanho do vetor a ser lido e uma segunda linha, contendo N elementos inteiros separados entre si por um espaço.

Saída

Para cada caso de teste, o programa deve imprimir um número inteiro correspondente ao número de elementos que ocorrem apenas uma vez no vetor. Após o valor, o programa deve imprimir o caractere de quebra de linha.

Exemplo

Entrada
2
10
3 4 10 10 3 10 4 10 1 2
7
3 6 2 9 2 7 9
Saída
2
3

3 Positivos e negativos - 4,0 pontos

Considerando a entrada de valores inteiros, ordene estes valores segundo o seguinte critério:

- Os números positivos devem aparecer primeiro, ordenados em ordem crescente.
- Após os valores positivos, devem aparecer os valores negativos, ordenados em ordem decrescente.

Entrada

A primeira linha de entrada contém um único inteiro positivo $N(1 < N < 10^5)$ que corresponde ao número de linhas de entrada que vêm logo a seguir. As próximas N linhas conterão, cada uma delas, um valor inteiro.

Saída

Apresente todos os valores lidos na entrada segundo a regra descrita acima. Cada número deve ser impresso em uma linha. Após imprimir último número quebre a linha.

Exemplo

Entrada
10
4
-32
-34
543
-3456
654
567
-87
6789
98
Saída
4
98
543
567
654
6789
-32
-34
-87
-3456

4 Strcmp() Nerd - Questão opcional - 3,0 pontos

Nerdildo está aprendendo a programar usando strings na linguagem C. Ele quer criar a sua própria versão da função `strcmp`, a qual ele vai denominar `strcmpnerd()`. Ele quer que sua função tenha a mesma funcionalidade da função `strcmp()` encontrada em `string.h`. Apenas o nome da função deve ser diferente. Nerdildo então procurou na internet a especificação da função `strcmp()` para implementar a função `strcmpnerd()` com as mesmas características. Ele encontrou as seguintes informações:

Protótipo: `int strcmp(char *s1, char* s2);`

A função inicia comparando os primeiros caracteres das duas strings. Se ele sforem iguais entre si, a função compara o par seguinte de caracteres (um de cada string) até encontrar um par cujos caracteres diferem entre si ou encontrar um caractere nulo (`'\0'`). O valor retornado pela função depende do motivo do término da comparação e pode ser:

- `-1`, se os caracteres na posição i nas duas strings diferem entre si e $s1[i]$ possui código na tabela ASCII inferior ao código de $s2[i]$;
- `0`, se as duas strings apontadas por $s1$ e $s2$ forem iguais;
- `1`, se se os caracteres na posição i nas duas strings diferem entre si e $s1[i]$ possui código na tabela ASCII superior ao código de $s2[i]$;

O problema é que Nerdildo não consegue programar essa função e pede a você que escreva essa função para ele. Além de implementar a função `strcmpnerd()` seu programa deve ler vários casos de teste para mostrar que a função está implementada corretamente, de acordo com o especificado acima para a função `strcmp()`.

Entrada

A primeira linha da entrada contém o número $n, n > 0$ de casos de teste. Cada caso de teste contém duas linhas. A primeira linha corresponde à primeira string a ser comparada e a segunda linha corresponde à segunda string. Você pode assumir que toda string lida terá no máximo 499 caracteres.

Saída

Para cada caso de teste seu programa deve imprimir em uma linha um dos três possíveis valores: `0`, `-1`, ou `1`, correspondendo ao retorno da função `strcmpnerd()` ao comparar a primeira linha com a segunda do caso de teste correspondente na entrada.

IMPORTANTE

Essa questão além de ser avaliada pelo Sharif, será também avaliada pelo professor. A nota definitiva da questão será a dada pelo professor. A soluções que não implementarem a função, mas fizerem uso da função `strcmp()` terão nota ZERO, mesmo que o Sharif aceite a solução como correta.

Exemplo

Entrada
5 um caso de teste um Caso de teste outro caso de teste outro caso de testf mais um caso de teste mais um caso de teste Antonio Pereira Antonia Pereira Antonia Ferreira Antonio
Saída
1 -1 0 1 -1