UNIVERSIDADE ESTADUAL DO PIAUÍ – UESPI
CURSO DE BACHARELADO EM COMPUTAÇÃO
DISCIPLINA: PROGRAMAÇÃO I
PROFESSOR: EYDER RIOS

1ª AVALIAÇÃO
13/01/2022

ORIENTAÇÕES GERAIS:

1. Siga atentamente as especificações da tarefa quanto ao formato da entrada e saída de seu programa.
2. Não implemente nenhum recurso gráfico nas suas soluções (janelas, menus, etc), nem utilize qualquer rotina para limpar a tela ou posicionar o cursor.
3. Todas as tarefas têm o mesmo valor na correção.
4. As tarefas não estão ordenadas por ordem de dificuldade.
5. Não utilize arquivos para entrada ou saída nos programas. Todos os dados devem ser lidos da entrada padrão (tipicamente o teclado) e escritos na saída padrão (tipicamente a tela).
6. Procure resolver o problema de maneira eficiente. Na correção, eficiência também será levada em conta. As soluções serão testadas com outras entradas além das apresentadas como exemplo nas tarefas.
7. A implementação pode ser feitas em grupos de ATÉ 3 (três) alunos.
8. No início do código fonte **deve haver um comentário informando os nomes dos autores**, sob pena de nulidade da questão.
9. Os autores de cada implementação poderão ser chamados para explicar o código entregue.

ENTREGA:

- Os códigos produzidos devem ser entregues em **um único arquivo compactado** em formato ZIP.
- O arquivo compactado deverá ser enviado para o e-mail: **eyder@phb.uespi.br**
- O **assunto do e-mail** deverá iniciar com "PROG1 – AVAL1: " seguido dos primeiros nomes dos autores.
- A entrega deverá ser feita até as 18:00 do dia 14/01/2022 (sexta-feira).
- Códigos considerados copiados da Internet ou de outras equipes receberão NOTA ZERO, independente de quem seja(m) o(s) autor(es).

UNIVERSIDADE ESTADUAL DO PIAUÍ – UESPI
CURSO DE BACHARELADO EM COMPUTAÇÃO
DISCIPLINA: PROGRAMAÇÃO I
PROFESSOR: EYDER RIOS

PROBLEMA 1:

A maioria dos livros publicados atualmente possuem um código de identificação único conhecido como ISBN. O International Standard Book Number (ISBN) é normalmente uma sequência de 10 digitos decimais porém, em alguns casos, a letra maiúscula "X" pode aparecer como décimo dígito. Hífens são incluídos em diversas posições no ISBN com o objetivo de tornar sua leitura mais fácil, porém não possuem nenhuma outra finalidade.

De fato, somente os nove primeiros dígitos do ISBN são utilizados para identificar um livro. O décimo caractere serve como um dígito para verificar se os 9 dígitos anteriores estão corretos. Este dígito verificador é selecionado de forma que o valor calculado, conforme mostrado no algoritmo a seguir, seja divisível por 11. Em alguns casos, o dígito verificador pode precisar ser igual a 10 para garantir a divisibilidade por 11. Nestes casos um caractere especial foi definido pelos *designers* do ISBN para representar o 10, e esse é o papel desempenhado pelo "X".

O algoritmo usado para verificar um ISBN é relativamente simples. Duas somas, s_1 e s_2, são calculadas sobre os dígitos do ISBN. s_1 é a soma parcial dos dígitos do ISBN, e s_2 é a soma das somas parciais em s_1. O ISBN está correto se o valor final de s_2 for divisível por 11. Um exemplo irá esclarecer o procedimento. Considere o ISBN válido 0-13-162959-X. Primeiro, observe para o cálculo de s_1:

Dígitos do ISBN	**0**	**1**	**3**	**1**	**6**	**2**	**9**	**5**	**9**	**X**	(10)
s_1 (somas parciais)	0	1	4	5	11	13	22	27	36	46	

O cálculo de s_2 é feito pela computação do total das somas parciais no cálculo de s_1:

s_2 (soma das somas parciais)	0	1	5	10	21	34	56	83	119	165

O ISBN estará correto se o somatório das somas parciais (s_2) for divisível por 11.

FORMATO DA ENTRADA

Cada linha do arquivo de entrada conterá um ISBN candidato, talvez precedido e/ou seguido por espaços adicionais. Nenhuma linha conterá mais de 80 caracteres, mas o ISBN candidato pode conter caracteres ilegais e mais ou menos do que os 10 dígitos exigidos. ISBNs válidos podem incluir hifens em locais arbitrários. A entrada termina com um ISBN especial: 0-00000-000-0.

FORMATO DA SAÍDA

A saída deve incluir uma exibição do ISBN candidato (sem espaços) e uma declaração se ele está correto ou errado. O exemplo mostrado abaixo ilustra o formato de saída esperado.

Exemplo de entrada

```
0-89237-010-6
  0-8306-3637-4
    0-8306-3637-5
0-00000-000-0
```

Saída correspondente

```
0-89237-010-6 está correto.
0-8306-3637-4 está correto.
0-8306-3637-5 está incorreto.
```

UNIVERSIDADE ESTADUAL DO PIAUÍ – UESPI
CURSO DE BACHARELADO EM COMPUTAÇÃO
DISCIPLINA: PROGRAMAÇÃO I
PROFESSOR: EYDER RIOS

PROBLEMA 2:

Carlos, um calouro de engenharia, está desenvolvendo uma notação posicional original para representar números inteiros. Ele a chamou de "Um Método Curioso" (UMC para abreviar). A notação UMC usa os mesmos dígitos que a notação decimal, ou seja, de 0 a 9.

Para converter um número N de UMC para notação decimal, você deve adicionar k termos, onde k é o número de dígitos de N (na notação UMC). O valor do i-ésimo termo, correspondente ao i-ésimo dígito a_i, contando da direita para a esquerda, é $a_i \times i!$. Por exemplo, 719_{UMC} é equivalente a 53_{10}, já que $7 \times 3! + 1 \times 2! + 9 \times 1! = 53$.

Carlos acabou de começar a estudar a teoria dos números e provavelmente não sabe quais propriedades um sistema de numeração deveria ter, mas no momento ele está interessado apenas em converter um número de UMC para decimal. Você poderia ajudá-lo?

FORMATO DA ENTRADA

Cada caso de teste é fornecido em uma única linha que contém uma string não vazia de no máximo 5 dígitos, representando um número em notação ACM. A string não tem zeros à esquerda. O último caso de teste é seguido por uma linha contendo um zero.

FORMATO DA SAÍDA

Para cada caso de teste, imprima uma única linha contendo a representação decimal do correspondente Número UMC.

Exemplo de entrada	**Saída correspondente**
719	53
1	1
15	7
110	8
102	8
0	

UNIVERSIDADE ESTADUAL DO PIAUÍ – UESPI
CURSO DE BACHARELADO EM COMPUTAÇÃO
DISCIPLINA: PROGRAMAÇÃO I
PROFESSOR: EYDER RIOS

PROBLEMA 3:

Recentemente, uma equipe de exploração descobriu uma tribo perdida na selva amazônica. Os nativos falam uma linguagem muito simples e regular, e para melhorar a comunicação entre nativos e exploradores, você deve escrever um sistema analisador de verbos. Os primeiros contatos com os indígenas já permitiram a dedução das seguintes regras:

1. Todos os verbos são regulares.
2. Todos os verbos terminam com uma consoante seguida pelo sufixo "en".
3. Existem apenas três tempos verbais: passado, presente e futuro.
4. Para conjugar o verbo, os nativos substituem o sufixo "en" por outro, conforme tabela a seguir:

Pessoa	Pronome	Presente	Passado	Futuro
1ª	Eu	o	ei	ai
2ª	Tu	os	es	ais
3ª	Ele	a	e	i
4ª	Nós	om	em	aem
5ª	Vós	ons	est	aist
6ª	Eles	am	im	aim

O verbo "**campten**", por exemplo, possui a seguinte conjugação:

Pessoa	Pronome	Presente	Passado	Futuro
1ª	Eu	campto	camptei	camptai
2ª	Tu	camptos	camptes	camptais
3ª	Ele	campta	campte	campti
4ª	Nós	camptom	camptem	camptaem
5ª	Vós	camptons	camptest	camptaist
6ª	Eles	camptam	camptim	camptaim

O programa deve analisar uma palavra, verificar se é verbo ou não e, se for verbo, indicar a pessoa e o tempo do verbo original.

ENTRADA

Uma lista de palavras, uma por linha.

SAÍDA

Uma palavra analisada por linha, seguida do resultado da análise, conforme mostrado nos exemplos.

EXEMPLO

Exemplo de entrada	**Saída correspondente**
`casos`	`casos - verbo casen, presente, 2a pessoa`
`porre`	`porre - verbo porren, pretérito, 3a pessoa`
`corraem`	`corraem - verbo corren, futuro, 4a pessoa`
`picel`	`picel - não é um tempo verbal`
`oficina`	`oficina - verbo oficinen, presente, 3a pessoa`
`param`	`param - verbo paren, presente, 6a pessoa`

UNIVERSIDADE ESTADUAL DO PIAUÍ – UESPI
CURSO DE BACHARELADO EM COMPUTAÇÃO
DISCIPLINA: PROGRAMAÇÃO I
PROFESSOR: EYDER RIOS

PROBLEMA 4:

Uma sequência de *N* números inteiros (cada número maior que zero) é considerada *M*-alternada se a sequência for composta por *M* segmentos consecutivos: o primeiro segmento com um elemento; o segundo segmento com dois elementos; e assim por diante, onde o *M*-ésimo segmento com *M* elementos. Além disso, a sequência deve satisfazer as seguintes condições:

1. os elementos de um determinado segmento devem ser todos ímpares ou todos pares;
2. se os elementos em um segmento são todos pares, então os elementos no segmento seguinte devem ser todos ímpares, e vice-versa.

Escreva um programa que, dado um inteiro $N > 1$ e uma sequência de *N* inteiros positivos, verifique se a sequência é *M*-alternada. O programa deve computar o valor para *M* se a sequência for *M*-alternada, ou produzir a string "NAO" se a sequência não for *M*-alternada.

FORMATO DA ENTRADA

A primeira linha informa o valor de *N*. A linha seguinte contém os *N* inteiros correspondentes à sequência a ser analisada. O arquivo de entrada contém várias instâncias (casos de teste) que terminam com uma linha contendo apenas um "%" (símbolo de porcentagem).

FORMATO DA SAÍDA

Cada linha de saída deve conter o valor de *M* ou a palavra "NAO", de acordo com as condições descritas acima. O arquivo de saída deve terminar cada saída correspondente a cada instância de entrada por uma linha com um "%" (símbolo de porcentagem).

EXEMPLO

Exemplo de entrada	**Saída correspondente**
10 12 3 7 2 10 4 5 13 5 11 % 8 11 2 3 4 5 77 33 44 %	4 % NAO %