

Universidade Federal de Goiás
Introdução à Programação
Exercícios sobre Funções Recursivas

Prof. Edmundo Sérgio Spoto Prof. Msc. Elias Batista Ferreira
Prof. Dr. Gustavo Teodoro Laureano
Profa. Dra. Luciana Berretta
Prof. Dr. Thierson Couto Rosa

Sumário

1	Quantas Vezes ‘x’ Ocorre?	2
2	Quantas Vezes “hi” Ocorre?	3
3	Quantidade de Cincos	4
4	Soma Recursiva	5
5	Soma Recursiva de Elementos de um Vetor	6
6	Coelhos Mutantes	7
7	Logaritmo na Base Dois Recursivo	8
8	Palíndromo Recursivo	9
9	Potência Recursiva	10
10	Prefixo Recursivo	11
11	Produto Recursivo	12
12	Próximo Elemento 10 Vezes o Atual	13

1 Quantas Vezes 'x' Ocorre?



(+)

Crie uma função recursiva que receba como entrada uma string e também o seu tamanho. Essa função deverá retornar quantas vezes o caractere 'x' aparece na string.

Entrada

A primeira linha na entrada contém o número n de casos de teste. Em seguida ocorrem n linhas, cada uma contendo uma string de tamanho máximo igual 5000 caracteres

Saída

A saída é formada por n linhas, cada uma contendo um número inteiro que indica o número de vezes que a letra 'x' ocorre na entrada. Todas as strings estão escritas em letras minúsculas.

Exemplo

Entrada
3 abra cadabra taxi xadrez bacaxi taxa caixa puxa lixa xarope xxx
Saída
0 7 4

2 Quantas Vezes “hi” Ocorre?



(+)

Crie uma função recursiva que receba como entrada uma string e também o seu tamanho. Essa função deverá retornar quantas vezes o caractere a subcadeia “hi” aparece na string.

Entrada

A primeira linha na entrada contém o número n de casos de teste. Em seguida ocorrem n linhas, cada uma contendo uma string de tamanho máximo igual 5000 caracteres

Saída

A saída é formada por n linhas, cada uma contendo um número inteiro que indica o número de vezes que a cadeia “hi” ocorre na string de entrada. Todas as strings estão escritas em letras minúsculas.

Exemplo

Entrada
4
hipotenusa hipotermia hilux hifi
hi
hihihi
xavante he
Saída
4
1
3
0

3 Quantidade de Cincos



(+)

Crie uma função recursiva que tenha como entrada números inteiros, e verifique quantidade de dígito 5 que aparece em cada número.

Entrada

A primeira linha na entrada contém um número inteiro n que corresponde ao número de casos de teste. Em seguida, há uma linha para cada um dos n casos de teste. Cada linha contém um número inteiro positivo.

Saída

A saída é formada por n linhas cada uma contendo um número inteiro que corresponde ao número de vezes em que um dígito 5 ocorre no caso de teste correspondente na entrada.

Exemplo

Entrada
5
505505550
589559955
90000000
35
5
Saída
6
5
0
1
1

4 Soma Recursiva



(+) Crie uma função recursiva que calcule a soma de 1 à n , onde n é um número inteiro positivo.

Entrada

A a primeira linha da entrada contém um número inteiro T que indica o número de casos de teste. Em seguida há T linhas, cada uma contendo um número inteiro positivo

Saída

A saída é formada por T linhas, cada uma contendo um número inteiro que representa a soma de 1 a n de uma linha correspondente na entrada.

Exemplo

Entrada
3
1
45
3
Saída
1
1035
6

5 Soma Recursiva de Elementos de um Vetor



(+)

Calcule a soma de todos os elementos de um dado vetor de inteiros usando uma função recursiva.

Entrada

A primeira linha da entrada contém o número de casos de teste. Cada caso de teste é formado por duas linhas. A primeira, contém o tamanho n ($1 \leq n \leq 30$) de um vetor. A segunda, contém n valores inteiros, separados entre si por um espaço.

Saída

A saída para cada caso de teste é formada por n linhas, cada uma contém um número inteiro que é a soma dos elementos do vetor correspondente na entrada.

Exemplo

Entrada
3
5
3 2 1 4 5
5
1 2 3 4 5
10
1 2 3 4 5 6 7 8 9 100
Saída
15
15
145

6 Coelhos Mutantes



(++)

Temos coelhos posicionados em uma linha, numerados de 1 a x . Os coelhos em posições ímpares possuem 2 orelhas, já os coelhos em posições pares possuem 3 orelhas, por terem sofrido mutações genéticas. Recursivamente retorne o número de orelhas de todos os coelhos enfileirados. (Sem estruturas de repetição ou multiplicação).

Entrada

A primeira linha na entrada contém um número inteiro n que corresponde ao número de casos de teste. Cada caso de teste contém o limite superior x do intervalo fechado $[1, x]$ que corresponde às posições onde há coelhos.

Saída

A saída é formada por n linhas cada uma contendo a soma do número de orelhas de coelhos de um caso de teste.

Exemplo

Entrada
4
1
3
5
7
Saída
2
7
12
17

7 Logaritmo na Base Dois Recursivo



(++)

O **piso** de um número real x é o único número inteiro i tal que $i \leq x \leq i + 1$. O piso de x é denotado por $\lfloor x \rfloor$. Em computação usa-se muito computar o piso do logaritmo na base 2 de um número x , isto é, $\lfloor \log_2 x \rfloor$. Lembrando que $\log_2 x = y$, se $2^y = x$. Por exemplo, a tabela abaixo mostra alguns números positivos e os respectivos pisos dos seus logaritmos na base 2:

x	$\lfloor \log_2 x \rfloor$
15	3
16	4
31	4
32	5
63	5
64	6

O cálculo de $\lfloor \log_2 x \rfloor$ pode ser obtido, gerando-se uma série de números inteiros resultantes de sucessivas divisões inteiras por dois, iniciado-se com o quociente de x dividido por dois e continuando-se com as divisões por dois dos sucessivos quocientes resultantes, até que se obtenha um quociente igual a um. O número de elementos na série resultante corresponde ao $\lfloor \log_2 x \rfloor$. Veja alguns exemplos:

$$\begin{aligned}\lfloor \log_2 15 \rfloor &= |\{7, 3, 1\}| = 3 \\ \lfloor \log_2 16 \rfloor &= |\{8, 4, 2, 1\}| = 4 \\ \lfloor \log_2 31 \rfloor &= |\{15, 7, 3, 1\}| = 4 \\ \lfloor \log_2 32 \rfloor &= |\{16, 8, 4, 2, 1\}| = 5 \\ \lfloor \log_2 63 \rfloor &= |\{31, 15, 7, 3, 1\}| = 5 \\ \lfloor \log_2 64 \rfloor &= |\{32, 16, 8, 4, 2, 1\}| = 6\end{aligned}$$

Escreva uma função recursiva para computar o piso do logaritmo na base dois de um número x fornecido como entrada. Lembre-se que $\lfloor \log_2 1 \rfloor = \log_2 1 = 0$.

Entrada

A primeira linha da entrada contém o número n ($0 < n \leq 100$) de casos de teste. Cada caso de teste é formado por uma linha contendo um único número inteiro positivo x para o qual deseja-se coputar $\lfloor \log_2 x \rfloor$.

Saída

Para cada caso de teste o seu programa deve imprimir uma linha contendo o valor do piso do logaritmo na base dois do número que corresponde ao caso de teste.

Exemplo

Entrada:
6
15
16
31
32
63
64

Saída:
3
4
4
5
5
6

8 Palíndromo Recursivo



(++)

Um palíndromo é uma palavra ou sequência de caracteres que tem a propriedade de poder ser lida tanto da esquerda para direita como da direita para a esquerda. Por exemplo, as seguintes palavras são palíndromos: *Ana*, *Bob*, *Otto*, *Mussum*. Escreva um programa que leia várias sequências de letras e que para cada uma delas chame uma função RECURSIVA que indique se uma sequência forma ou não um palíndromo.

Entrada

A entrada é formada por uma linha inicial que contém um número inteiro que indica o número de casos de testes. Para cada caso de teste há duas linhas. A primeira contém um número que indica o tamanho da sequência de letras a ser lida. O tamanho máximo da sequência é 2.000 letras. A segunda linha contém a sequência de letras, sendo que entre uma letra e outra há um espaço. A última letra da sequência vem seguida do caractere de quebra de linha.

Saída

Para cada sequência da entrada o seu programa deve emitir uma das seguintes respostas: PALINDROMO, se a sequência for um palíndromo, ou NAO PALINDROMO, em caso contrário. Após cada palavra impressa deve haver apenas um caractere de quebra de linha.

Exemplos

Entrada:
5
5
N r C q K
4
M S K Y
3
L t V
2
K K
1
J

Saída:
NAO PALINDROMO
NAO PALINDROMO
NAO PALINDROMO
PALINDROMO
PALINDROMO

Observação

Cuidado ao ler a sequência de caracteres. Se você utilizar `scanf("%c", &vet[j])`, dentro do comando `while` para a leitura, o programa vai ler o espaços também e armazena-los no vetor. Sugestão: `while (j<tam-1) { {scanf("%c " &vet[j]); scanf("%c" &vet[j]) } getchar()}. Repare que o scanf() dentro do while contém um espaço após %c e que o scanf() fora do comando while não tem esse espaço.`

9 Potência Recursiva



(++)

Dados dois números inteiros positivos M e N , escreva uma função recursiva que calcule o valor de M^N . Lembre-se que $M^0 = 1$.

Entrada

A primeira linha da entrada contém o número T , $1 \leq T \leq 300$ de casos de teste. Cada caso de teste é formado por uma linha, contendo dois números inteiros positivos separados por um espaço.

Saída

Para cada caso de teste, seu programa deve imprimir a potência cuja base é o primeiro número e o expoente o segundo número.

Exemplo

Entrada:	Saída:
3	32
2 5	19683
3 9	1
1 50	

10 Prefixo Recursivo



(++)

Um prefixo de uma cadeia de caracteres é qualquer sub-cadeia que possa ser obtida apagando zero ou mais caracteres da extremidade direita da cadeia original. Escreva uma função RECURSIVA que receba como parâmetro o endereço de memória onde está armazenada uma cadeia de caracteres e imprima todos os prefixos válidos da cadeia, exceto a cadeia vazia.

Escreva um programa para ler várias cadeias de caracteres da entrada e imprimir para cada cadeia o conjunto de prefixos não vazios da cadeia, utilizando a função mencionada acima.

Entrada

A primeira linha da entrada contém o número C , $1 \leq C \leq 300$ de casos de testes. As C linhas seguintes contêm, cada uma, um caso de teste formado por uma cadeia de caracteres.

Saída

Para cada caso de teste da entrada seu programa deve imprimir uma linha contendo a frase : *Caso de teste #*. Em seguida deve imprimir todos os prefixos não vazios da cadeia correspondente ao caso de teste. Deve ser impresso um prefixo por linha e a primeira linha deve corresponder ao maior prefixo. A segunda linha deve conter o segundo maior prefixo, e assim por diante.

Exemplo

Entrada:
3
oi ola
casa
mesa

Saída:
Caso de teste 1
oi ola
oi ol
oi o
oi
oi
o
Caso de teste 2
casa
cas
ca
c
Caso de teste 3
mesa
mes
me
m

11 Produto Recursivo



(++)

Dados dois números inteiros positivos M e N , o produto de M por N é igual a zero se um dos dois é zero, caso contrário, o produto pode ser definido como N somas de M por ele mesmo. Escreva uma função recursiva que calcule o produto de um número inteiro por outro, utilizando somas.

Entrada

A primeira linha da entrada contém o número T , $1 \leq T \leq 300$ de casos de teste. Cada caso de teste é formado por uma linha, contendo dois números inteiros positivos separados por um espaço.

Saída

Para cada caso de teste, seu programa deve imprimir o produto entre os dois números que formam o caso de teste.

Exemplo

Entrada:	Saída:
3	253
23 11	2250
45 50	0
0 900	

12 Próximo Elemento 10 Vezes o Atual



(+++)

Crie uma função recursiva que receba um vetor de inteiros e o seu respectivo tamanho como parâmetros. Essa função deverá verificar se há algum valor nesse vetor que, multiplicado por 10, seja igual ao próximo elemento, e então retornar 1. Caso contrário, retornar 0.

Entrada

A primeira linha da entrada contém o número de casos de teste. Cada caso de teste é formado por duas linhas. A primeira, contém o tamanho n ($1 \leq n \leq 30$) de um vetor. A segunda, contém n valores inteiros, separados entre si por um espaço.

Saída

A saída para cada caso de teste é formada por n linhas, cada uma contém uma das palavras: VERDADEIRO, se a condição é satisfeita pelo menos uma vez no vetor, ou FALSO, em caso contrário.

Exemplo

Entrada
3
6
0 10 100 1000 10000 100000
4
20 30 3 30
5
1 20 2 30 4
Saída
VERDADEIRO
VERDADEIRO
FALSO