

Iniciado em	quinta, 16 set 2021, 18:58
Estado	Finalizada
Concluída em	sábado, 18 set 2021, 18:56
Tempo empregado	1 dia 23 horas
Avaliar	10,00 de um máximo de 10,00(100%)

Aviso!

?

Questão 1

Correto

Atingiu 1,00 de 1,00

Média aritmética

Emidio está dando um curso de "como ser um Emidio". Ele acredita que precisa fazer avaliações rigorosas, para ter certeza de que seus alunos estão aprendendo tudo corretamente. Para ter uma ideia de como os alunos estão se saindo, ele resolveu calcular a média da primeira prova. O problema é que Emidio não se deu ao trabalho de contar quantos alunos estão fazendo o seu curso.

Já sabe, né? O Emidio é o Emidio... É claro que ele pediu a sua ajuda para esta tarefa.

Entrada

A entrada é composta por um número $2 \leq n \leq 10^6$ desconhecido de linhas. Cada linha indica a nota $0 \leq x \leq 10^3$ de um aluno, exceto a última, que contém somente o número -1 para indicar que não há mais notas a serem lidas.

Saída

Escreva um [programa](#) que leia a entrada e imprima a parte inteira da média aritmética dos alunos de Emidio.

Notas

O valor -1 não deve ser considerado para a média.

For example:

Input	Result
9 7 10 -1	8
5 -1	5

Aviso!

?

Input	Result
4	24
5	
12	
4	
99	
-1	

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 notas = []
2 entrada = float(input())
3
4 while entrada != -1:
5     notas.append(entrada)
6     entrada = float(input())
7
8 def media(lista):
9     media = 0
10    for x in lista:
11        media+= x
12    return media/len(lista)
13
14
15 print(int(media(notas)))

```

	Input	Expected	Got	
✓	9 7 10 -1	8	8	✓

Aviso!

?

	Input	Expected	Got	
✓	5 -1	5	5	✓
✓	4 5 12 4 99 -1	24	24	✓
✓	802 765 992 1 521 220 380 729 969 184 887 104 641 909 378 724 582 387 583 241 294 159 198 653 369 418 692 36 001 5 3	478	478	✓

Aviso!

?

	Input	Expected	Got	
	703			
	971			
	304			
	394			
	491			
	525			
	464			
	219			
	183			
	648			
	796			
	287			
	979			
	395			
	356			
	702			
	667			
	743			
	976			
	908			
	728			
	134			
	106			
	380			
	193			
	214			
	71			
	920			
	114			
	587			
	543			
	817			
	248			
	537			
	901			
	739			
	752			
	364			
	519			
	5			
	702			

Aviso!

?

	Input	Expected	Got	
	444			
	913			
	681			
	529			
	959			
	72			
	196			
	392			
	738			
	103			
	119			
	872			
	900			
	189			
	65			
	113			
	260			
	985			
	228			
	537			
	217			
	735			
	785			
	445			
	636			
	214			
	196			
	690			
	553			
	822			
	392			
	687			
	425			
	763			
	216			
	73			
	525			
	412			
	175			
	3			
	205			

Aviso!

?

	Input	Expected	Got	
	965			
	825			
	105			
	153			
	580			
	218			
	103			
	255			
	136			
	641			
	472			
	872			
	115			
	607			
	197			
	19			
	494			
	577			
	572			
	5			
	659			
	259			
	120			
	421			
	165			
	194			
	637			
	577			
	39			
	900			
	472			
	4			
	724			
	577			
	157			
	304			
	486			
	261			
	219			
	2			
	522			

Aviso!

?

	Input	Expected	Got	
	411			
	874			
	397			
	18			
	70			
	417			
	512			
	338			
	679			
	517			
	997			
	938			
	328			
	418			
	793			
	522			
	745			
	59			
	561			
	644			
	532			
	255			
	57			
	108			
	413			
	51			
	284			
	364			
	300			
	597			
	646			
	712			
	470			
	42			
	730			
	231			
	459			
	932			
	559			
	9			
	108			

Aviso!

?

	Input	Expected	Got	
	565			
	766			
	466			
	673			
	559			
	678			
	417			
	618			
	930			
	751			
	840			
	184			
	809			
	639			
	287			
	550			
	923			
	341			
	851			
	209			
	987			
	252			
	370			
	29			
	672			
	601			
	178			
	603			
	860			
	6			
	431			
	114			
	463			
	588			
	788			
	712			
	956			
	895			
	10			
	5			
	556			

Aviso!

?

	Input	Expected	Got	
	860			
	760			
	835			
	498			
	737			
	384			
	110			
	78			
	925			
	320			
	755			
	176			
	690			
	784			
	848			
	981			
	653			
	140			
	840			
	659			
	262			
	954			
	812			
	850			
	431			
	523			
	495			
	16			
	233			
	70			
	352			
	92			
	520			
	877			
	280			
	256			
	260			
	390			
	5			
	100			

Aviso!

?

	Input	Expected	Got	
	780			
	51			
	89			
	253			
	900			
	760			
	906			
	730			
	599			
	565			
	992			
	243			
	66			
	531			
	364			
	590			
	26			
	380			
	823			
	787			
	422			
	605			
	306			
	298			
	885			
	562			
	249			
	965			
	277			
	124			
	365			
	56			
	175			
	144			
	309			
	765			
	595			
	215			
	105			
	4			
	0			

Aviso!

?

	Input	Expected	Got	
	176			
	126			
	536			
	398			
	181			
	816			
	114			
	251			
	328			
	901			
	674			
	933			
	206			
	662			
	507			
	458			
	601			
	162			
	735			
	725			
	217			
	481			
	591			
	51			
	791			
	355			
	646			
	696			
	540			
	530			
	165			
	717			
	346			
	391			
	114			
	527			
	898			
	228			
	779			
	5			
	829			

Aviso!

?

	Input	Expected	Got	
	142 849 24 494 45 172 95 207 908 510 424 78 100 166 869 456 812 254 996 341 109 402 688 501 -1			
✓	815 845 838 738 411 47 309 7 142 835 139 86 656 0 3	496	496	✓

Aviso!

?

	Input	Expected	Got	
	612			
	146			
	504			
	751			
	613			
	181			
	335			
	370			
	23			
	602			
	864			
	974			
	778			
	484			
	143			
	661			
	299			
	678			
	188			
	727			
	89			
	927			
	35			
	96			
	68			
	560			
	926			
	154			
	216			
	395			
	502			
	518			
	232			
	697			
	959			
	535			
	568			
	293			
	506			
	2			
	506			

Aviso!

?

	Input	Expected	Got	
	459			
	255			
	363			
	944			
	88			
	714			
	933			
	767			
	593			
	659			
	546			
	519			
	384			
	332			
	587			
	945			
	948			
	431			
	851			
	32			
	624			
	58			
	955			
	320			
	16			
	180			
	578			
	1000			
	776			
	860			
	585			
	234			
	806			
	638			
	868			
	894			
	352			
	490			
	579			
	5			
	178			

Aviso!

?

	Input	Expected	Got	
	586			
	153			
	223			
	608			
	430			
	167			
	555			
	861			
	708			
	278			
	174			
	766			
	923			
	494			
	782			
	102			
	763			
	781			
	878			
	622			
	56			
	803			
	427			
	694			
	670			
	11			
	45			
	160			
	51			
	680			
	999			
	638			
	523			
	221			
	245			
	953			
	78			
	491			
	504			
	5			
	129			

Aviso!

?

	Input	Expected	Got	
	678			
	241			
	381			
	863			
	713			
	484			
	625			
	184			
	51			
	937			
	240			
	854			
	54			
	624			
	524			
	65			
	670			
	374			
	116			
	39			
	372			
	444			
	563			
	284			
	380			
	205			
	52			
	561			
	399			
	529			
	19			
	768			
	460			
	90			
	320			
	173			
	264			
	635			
	17			
	5			
	371			

Aviso!

?

	Input	Expected	Got	
	978			
	169			
	625			
	601			
	383			
	380			
	961			
	757			
	187			
	0			
	820			
	631			
	253			
	103			
	10			
	148			
	846			
	571			
	238			
	374			
	280			
	5			
	835			
	371			
	15			
	698			
	635			
	650			
	745			
	951			
	911			
	412			
	811			
	226			
	12			
	193			
	606			
	974			
	871			
	3			
	804			

Aviso!

?

	Input	Expected	Got	
	460			
	114			
	607			
	253			
	124			
	755			
	98			
	386			
	993			
	473			
	356			
	688			
	998			
	727			
	703			
	695			
	52			
	42			
	129			
	693			
	954			
	541			
	503			
	179			
	553			
	387			
	475			
	216			
	337			
	268			
	570			
	487			
	382			
	176			
	740			
	196			
	932			
	838			
	272			
	4			

Aviso!

?

	Input	Expected	Got	
	629			
	993			
	998			
	45			
	695			
	692			
	97			
	738			
	821			
	481			
	691			
	51			
	674			
	560			
	295			
	60			
	34			
	201			
	87			
	993			
	772			
	574			
	64			
	948			
	313			
	261			
	569			
	842			
	533			
	874			
	842			
	852			
	866			
	530			
	898			
	250			
	221			
	995			
	208			
	3			
	205			

Aviso!

?

	Input	Expected	Got	
	368 474 840 928 769 900 653 971 678 645 742 251 710 379 255 -1			
✓	275 12 986 292 981 137 477 198 356 270 369 29 130 395 580 700 533 830 498 639 198 823 5	503	503	✓

Aviso!

?

	Input	Expected	Got	
	525			
	496			
	517			
	652			
	814			
	778			
	61			
	88			
	790			
	46			
	71			
	770			
	874			
	548			
	659			
	230			
	508			
	27			
	950			
	638			
	112			
	220			
	338			
	645			
	49			
	836			
	975			
	248			
	348			
	970			
	253			
	564			
	466			
	460			
	215			
	279			
	237			
	967			
	50			
	703			

Aviso!

?

	Input	Expected	Got	
	129			
	486			
	577			
	367			
	144			
	497			
	875			
	862			
	446			
	513			
	665			
	666			
	541			
	1000			
	406			
	66			
	974			
	654			
	415			
	944			
	597			
	669			
	409			
	56			
	574			
	378			
	293			
	540			
	126			
	9			
	933			
	946			
	495			
	509			
	312			
	330			
	5			
	187			
	222			
	2			
	100			

Aviso!

?

	Input	Expected	Got	
	546			
	498			
	931			
	546			
	904			
	997			
	519			
	247			
	101			
	462			
	844			
	770			
	561			
	590			
	343			
	940			
	883			
	573			
	65			
	892			
	506			
	11			
	77			
	705			
	323			
	97			
	401			
	200			
	979			
	543			
	590			
	525			
	731			
	520			
	761			
	325			
	517			
	279			
	772			
	3			
	72			

Aviso!

?

	Input	Expected	Got	
	106			
	78			
	683			
	696			
	421			
	622			
	269			
	685			
	378			
	851			
	190			
	389			
	618			
	895			
	402			
	715			
	295			
	603			
	694			
	528			
	883			
	909			
	950			
	403			
	669			
	274			
	610			
	638			
	536			
	918			
	760			
	332			
	996			
	443			
	719			
	107			
	64			
	678			
	702			
	2			
	528			

Aviso!

?

	Input	Expected	Got	
	672			
	521			
	837			
	256			
	614			
	551			
	242			
	907			
	244			
	460			
	789			
	843			
	409			
	191			
	511			
	373			
	491			
	149			
	600			
	409			
	599			
	622			
	94			
	41			
	340			
	201			
	797			
	17			
	683			
	929			
	236			
	44			
	140			
	72			
	992			
	754			
	313			
	233			
	550			
	3			
	33			

Aviso!

?

	Input	Expected	Got	
	448			
	400			
	793			
	330			
	602			
	856			
	511			
	441			
	455			
	920			
	39			
	77			
	14			
	772			
	107			
	906			
	568			
	816			
	279			
	186			
	742			
	13			
	326			
	814			
	4			
	770			
	817			
	928			
	119			
	374			
	311			
	568			
	465			
	103			
	588			
	757			
	959			
	98			
	107			
	4			
	709			

Aviso!

?

	Input	Expected	Got	
	927			
	872			
	413			
	698			
	979			
	8			
	956			
	484			
	287			
	142			
	225			
	992			
	158			
	729			
	996			
	929			
	546			
	614			
	738			
	610			
	925			
	305			
	74			
	27			
	583			
	521			
	676			
	372			
	718			
	470			
	80			
	645			
	31			
	183			
	32			
	701			
	191			
	989			
	104			
	9			
	821			

Aviso!

?

	Input	Expected	Got	
	410			
	160			
	979			
	829			
	846			
	597			
	374			
	459			
	335			
	985			
	383			
	330			
	749			
	100			
	604			
	270			
	467			
	976			
	678			
	937			
	746			
	12			
	969			
	929			
	45			
	359			
	810			
	33			
	543			
	979			
	854			
	643			
	138			
	522			
	472			
	675			
	810			
	846			
	123			
	4			
	520			

Aviso!

?

	Input	Expected	Got	
	207			
	164			
	269			
	998			
	458			
	229			
	464			
	123			
	907			
	91			
	869			
	920			
	750			
	487			
	965			
	108			
	988			
	688			
	651			
	966			
	231			
	294			
	795			
	753			
	-1			

Passou em todos os teste! ✓

Question author's solution (Python3):

```
1 sum = 0
2 cnt = 0
3 while True:
4     x = int(input())
5     if x == -1:
6         break
7     sum += x
```

Aviso!

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 2

Correto

Atingiu 1,00 de 1,00

Conversão de Real para Dólar

Nem mesmo a intervenção do Banco Central nesta terça evitou que a moeda americana registrasse sua quarta alta seguida na [bolsa de valores](#).

Sua casa de câmbio vende dólares em lotes, por uma módica taxa adicional de 2,5% por lote (devido a contratos de ativos futuros).

Entrada

A entrada consiste em uma linha com um valor real positivo $0 \leq n$, representando a atual cotação do dólar. A linha seguinte apresenta um valor inteiro positivo $0 \leq l$ indicando o tamanho do lote (quantos dólares serão comprados). A última linha apresenta um valor inteiro positivo indicando a quantidade de lotes $0 \leq q$ a serem vendidos.

Saída

Apresente, para cada lote, uma linha contendo a mensagem indicando o número do lote e seu custo para o cliente, com duas casas decimais, conforme os exemplos.

Observações

- No primeiro caso de teste, a cotação do dólar é de 4.20, serão comprados 1000 dólares em cada lote, e serão comprados 2 lotes. O valor total de cada lote é então de 4305.00 reais.
- No segundo caso de teste, a cotação do dólar é de 4.50, serão comprados 10000 dólares em cada lote, e serão comprados 4 lotes. O valor total de cada lote é então de 46125.00 reais.
- No terceiro caso de teste, a cotação do dólar é de 4.50, serão comprados 10000 dólares em cada lote, e serão comprados 3 lotes. O valor total de cada lote é então de 46125.00 reais.

For example:

Input	Result
4.20	Lote: 1 - Total da venda: R\$ 4305.00
1000	Lote: 2 - Total da venda: R\$ 4305.00
2	

Aviso!

?



Input	Result
4.50	Lote: 1 - Total da venda: R\$ 46125.00
10000	Lote: 2 - Total da venda: R\$ 46125.00
4	Lote: 3 - Total da venda: R\$ 46125.00
	Lote: 4 - Total da venda: R\$ 46125.00
4.50	Lote: 1 - Total da venda: R\$ 46125.00
10000	Lote: 2 - Total da venda: R\$ 46125.00
3	Lote: 3 - Total da venda: R\$ 46125.00

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 |
2 | TAXA = 0.025
3 |
4 | cotacao = float(input())
5 | tamanho_lote = int(input())
6 | qtd_lotes = int(input())
7 |
8 | custo_lote = cotacao * tamanho_lote * (1 + TAXA)
9 |
10 | for x in range(1,qtd_lotes+1):
11 |     print(f"Lote: {x} - Total da venda: R$ {custo_lote:.2f}")

```

	Input	Expected	Got	
	4.20 1000	Lote: 1 - Total da venda: R\$ 4305.00 Lote: 2 - Total da venda: R\$ 4305.00	Lote: 1 - Total da venda: R\$ 4305.00 Lote: 2 - Total da venda: R\$ 4305.00	

Aviso!

?

	Input	Expected	Got	
✓	4.50 10000 4	Lote: 1 - Total da venda: R\$ 46125.00 Lote: 2 - Total da venda: R\$ 46125.00 Lote: 3 - Total da venda: R\$ 46125.00 Lote: 4 - Total da venda: R\$ 46125.00	Lote: 1 - Total da venda: R\$ 46125.00 Lote: 2 - Total da venda: R\$ 46125.00 Lote: 3 - Total da venda: R\$ 46125.00 Lote: 4 - Total da venda: R\$ 46125.00	✓
✓	4.50 10000 3	Lote: 1 - Total da venda: R\$ 46125.00 Lote: 2 - Total da venda: R\$ 46125.00 Lote: 3 - Total da venda: R\$ 46125.00	Lote: 1 - Total da venda: R\$ 46125.00 Lote: 2 - Total da venda: R\$ 46125.00 Lote: 3 - Total da venda: R\$ 46125.00	✓
✓	3.90 5000 1	Lote: 1 - Total da venda: R\$ 19987.50	Lote: 1 - Total da venda: R\$ 19987.50	✓
✓	0.01 10000 15	Lote: 1 - Total da venda: R\$ 102.50 Lote: 2 - Total da venda: R\$ 102.50 Lote: 3 - Total da venda: R\$ 102.50 Lote: 4 - Total da venda: R\$ 102.50 Lote: 5 - Total da venda: R\$ 102.50 Lote: 6 - Total da venda: R\$ 102.50 Lote: 7 - Total da venda: R\$ 102.50 Lote: 8 - Total da venda: R\$ 102.50 Lote: 9 - Total da venda: R\$ 102.50 Lote: 10 - Total da venda: R\$ 102.50 Lote: 11 - Total da venda: R\$ 102.50 Lote: 12 - Total da venda: R\$ 102.50 Lote: 13 - Total da venda: R\$ 102.50 Lote: 14 - Total da venda: R\$ 102.50 Lote: 15 - Total da venda: R\$ 102.50	Lote: 1 - Total da venda: R\$ 102.50 Lote: 2 - Total da venda: R\$ 102.50 Lote: 3 - Total da venda: R\$ 102.50 Lote: 4 - Total da venda: R\$ 102.50 Lote: 5 - Total da venda: R\$ 102.50 Lote: 6 - Total da venda: R\$ 102.50 Lote: 7 - Total da venda: R\$ 102.50 Lote: 8 - Total da venda: R\$ 102.50 Lote: 9 - Total da venda: R\$ 102.50 Lote: 10 - Total da venda: R\$ 102.50 Lote: 11 - Total da venda: R\$ 102.50 Lote: 12 - Total da venda: R\$ 102.50 Lote: 13 - Total da venda: R\$ 102.50 Lote: 14 - Total da venda: R\$ 102.50 Lote: 15 - Total da venda: R\$ 102.50	✓
✓	1 1 1000	Lote: 1 - Total da venda: R\$ 1.02 Lote: 2 - Total da venda: R\$ 1.02 Lote: 3 - Total da venda: R\$ 1.02 Lote: 4 - Total da venda: R\$ 1.02 Lote: 5 - Total da venda: R\$ 1.02 Lote: 6 - Total da venda: R\$ 1.02 Lote: 7 - Total da venda: R\$ 1.02 Lote: 8 - Total da venda: R\$ 1.02 Lote: 9 - Total da venda: R\$ 1.02 Lote: 10 - Total da venda: R\$ 1.02 Lote: 11 - Total da venda: R\$ 1.02 Lote: 12 - Total da venda: R\$ 1.02	Lote: 1 - Total da venda: R\$ 1.02 Lote: 2 - Total da venda: R\$ 1.02 Lote: 3 - Total da venda: R\$ 1.02 Lote: 4 - Total da venda: R\$ 1.02 Lote: 5 - Total da venda: R\$ 1.02 Lote: 6 - Total da venda: R\$ 1.02 Lote: 7 - Total da venda: R\$ 1.02 Lote: 8 - Total da venda: R\$ 1.02 Lote: 9 - Total da venda: R\$ 1.02 Lote: 10 - Total da venda: R\$ 1.02 Lote: 11 - Total da venda: R\$ 1.02 Lote: 12 - Total da venda: R\$ 1.02	✓

Aviso!

?

	Input	Expected	Got	
		Lote: 13 - Total da venda: R\$ 1.02 Lote: 14 - Total da venda: R\$ 1.02 Lote: 15 - Total da venda: R\$ 1.02 Lote: 16 - Total da venda: R\$ 1.02 Lote: 17 - Total da venda: R\$ 1.02 Lote: 18 - Total da venda: R\$ 1.02 Lote: 19 - Total da venda: R\$ 1.02 Lote: 20 - Total da venda: R\$ 1.02 Lote: 21 - Total da venda: R\$ 1.02 Lote: 22 - Total da venda: R\$ 1.02 Lote: 23 - Total da venda: R\$ 1.02 Lote: 24 - Total da venda: R\$ 1.02 Lote: 25 - Total da venda: R\$ 1.02 Lote: 26 - Total da venda: R\$ 1.02 Lote: 27 - Total da venda: R\$ 1.02 Lote: 28 - Total da venda: R\$ 1.02 Lote: 29 - Total da venda: R\$ 1.02 Lote: 30 - Total da venda: R\$ 1.02 Lote: 31 - Total da venda: R\$ 1.02 Lote: 32 - Total da venda: R\$ 1.02 Lote: 33 - Total da venda: R\$ 1.02 Lote: 34 - Total da venda: R\$ 1.02 Lote: 35 - Total da venda: R\$ 1.02 Lote: 36 - Total da venda: R\$ 1.02 Lote: 37 - Total da venda: R\$ 1.02 Lote: 38 - Total da venda: R\$ 1.02 Lote: 39 - Total da venda: R\$ 1.02 Lote: 40 - Total da venda: R\$ 1.02 Lote: 41 - Total da venda: R\$ 1.02 Lote: 42 - Total da venda: R\$ 1.02 Lote: 43 - Total da venda: R\$ 1.02 Lote: 44 - Total da venda: R\$ 1.02 Lote: 45 - Total da venda: R\$ 1.02 Lote: 46 - Total da venda: R\$ 1.02 Lote: 47 - Total da venda: R\$ 1.02 Lote: 48 - Total da venda: R\$ 1.02 Lote: 49 - Total da venda: R\$ 1.02 Lote: 50 - Total da venda: R\$ 1.02 Lote: 51 - Total da venda: R\$ 1.02 Lote: 52 - Total da venda: R\$ 1.02 Lote: 53 - Total da venda: R\$ 1.02	Lote: 13 - Total da venda: R\$ 1.02 Lote: 14 - Total da venda: R\$ 1.02 Lote: 15 - Total da venda: R\$ 1.02 Lote: 16 - Total da venda: R\$ 1.02 Lote: 17 - Total da venda: R\$ 1.02 Lote: 18 - Total da venda: R\$ 1.02 Lote: 19 - Total da venda: R\$ 1.02 Lote: 20 - Total da venda: R\$ 1.02 Lote: 21 - Total da venda: R\$ 1.02 Lote: 22 - Total da venda: R\$ 1.02 Lote: 23 - Total da venda: R\$ 1.02 Lote: 24 - Total da venda: R\$ 1.02 Lote: 25 - Total da venda: R\$ 1.02 Lote: 26 - Total da venda: R\$ 1.02 Lote: 27 - Total da venda: R\$ 1.02 Lote: 28 - Total da venda: R\$ 1.02 Lote: 29 - Total da venda: R\$ 1.02 Lote: 30 - Total da venda: R\$ 1.02 Lote: 31 - Total da venda: R\$ 1.02 Lote: 32 - Total da venda: R\$ 1.02 Lote: 33 - Total da venda: R\$ 1.02 Lote: 34 - Total da venda: R\$ 1.02 Lote: 35 - Total da venda: R\$ 1.02 Lote: 36 - Total da venda: R\$ 1.02 Lote: 37 - Total da venda: R\$ 1.02 Lote: 38 - Total da venda: R\$ 1.02 Lote: 39 - Total da venda: R\$ 1.02 Lote: 40 - Total da venda: R\$ 1.02 Lote: 41 - Total da venda: R\$ 1.02 Lote: 42 - Total da venda: R\$ 1.02 Lote: 43 - Total da venda: R\$ 1.02 Lote: 44 - Total da venda: R\$ 1.02 Lote: 45 - Total da venda: R\$ 1.02 Lote: 46 - Total da venda: R\$ 1.02 Lote: 47 - Total da venda: R\$ 1.02 Lote: 48 - Total da venda: R\$ 1.02 Lote: 49 - Total da venda: R\$ 1.02 Lote: 50 - Total da venda: R\$ 1.02 Lote: 51 - Total da venda: R\$ 1.02 Lote: 52 - Total da venda: R\$ 1.02 Lote: 53 - Total da venda: R\$ 1.02	
Aviso!				?

	Input	Expected	Got	
Aviso!		Lote: 54 - Total da venda: R\$ 1.02	Lote: 54 - Total da venda: R\$ 1.02	
		Lote: 55 - Total da venda: R\$ 1.02	Lote: 55 - Total da venda: R\$ 1.02	
		Lote: 56 - Total da venda: R\$ 1.02	Lote: 56 - Total da venda: R\$ 1.02	
		Lote: 57 - Total da venda: R\$ 1.02	Lote: 57 - Total da venda: R\$ 1.02	
		Lote: 58 - Total da venda: R\$ 1.02	Lote: 58 - Total da venda: R\$ 1.02	
		Lote: 59 - Total da venda: R\$ 1.02	Lote: 59 - Total da venda: R\$ 1.02	
		Lote: 60 - Total da venda: R\$ 1.02	Lote: 60 - Total da venda: R\$ 1.02	
		Lote: 61 - Total da venda: R\$ 1.02	Lote: 61 - Total da venda: R\$ 1.02	
		Lote: 62 - Total da venda: R\$ 1.02	Lote: 62 - Total da venda: R\$ 1.02	
		Lote: 63 - Total da venda: R\$ 1.02	Lote: 63 - Total da venda: R\$ 1.02	
		Lote: 64 - Total da venda: R\$ 1.02	Lote: 64 - Total da venda: R\$ 1.02	
		Lote: 65 - Total da venda: R\$ 1.02	Lote: 65 - Total da venda: R\$ 1.02	
		Lote: 66 - Total da venda: R\$ 1.02	Lote: 66 - Total da venda: R\$ 1.02	
		Lote: 67 - Total da venda: R\$ 1.02	Lote: 67 - Total da venda: R\$ 1.02	
		Lote: 68 - Total da venda: R\$ 1.02	Lote: 68 - Total da venda: R\$ 1.02	
		Lote: 69 - Total da venda: R\$ 1.02	Lote: 69 - Total da venda: R\$ 1.02	
		Lote: 70 - Total da venda: R\$ 1.02	Lote: 70 - Total da venda: R\$ 1.02	
		Lote: 71 - Total da venda: R\$ 1.02	Lote: 71 - Total da venda: R\$ 1.02	
		Lote: 72 - Total da venda: R\$ 1.02	Lote: 72 - Total da venda: R\$ 1.02	
		Lote: 73 - Total da venda: R\$ 1.02	Lote: 73 - Total da venda: R\$ 1.02	
		Lote: 74 - Total da venda: R\$ 1.02	Lote: 74 - Total da venda: R\$ 1.02	
		Lote: 75 - Total da venda: R\$ 1.02	Lote: 75 - Total da venda: R\$ 1.02	
		Lote: 76 - Total da venda: R\$ 1.02	Lote: 76 - Total da venda: R\$ 1.02	
		Lote: 77 - Total da venda: R\$ 1.02	Lote: 77 - Total da venda: R\$ 1.02	
		Lote: 78 - Total da venda: R\$ 1.02	Lote: 78 - Total da venda: R\$ 1.02	
		Lote: 79 - Total da venda: R\$ 1.02	Lote: 79 - Total da venda: R\$ 1.02	
		Lote: 80 - Total da venda: R\$ 1.02	Lote: 80 - Total da venda: R\$ 1.02	
		Lote: 81 - Total da venda: R\$ 1.02	Lote: 81 - Total da venda: R\$ 1.02	
		Lote: 82 - Total da venda: R\$ 1.02	Lote: 82 - Total da venda: R\$ 1.02	
		Lote: 83 - Total da venda: R\$ 1.02	Lote: 83 - Total da venda: R\$ 1.02	
		Lote: 84 - Total da venda: R\$ 1.02	Lote: 84 - Total da venda: R\$ 1.02	
		Lote: 85 - Total da venda: R\$ 1.02	Lote: 85 - Total da venda: R\$ 1.02	
		Lote: 86 - Total da venda: R\$ 1.02	Lote: 86 - Total da venda: R\$ 1.02	
		Lote: 87 - Total da venda: R\$ 1.02	Lote: 87 - Total da venda: R\$ 1.02	
		Lote: 88 - Total da venda: R\$ 1.02	Lote: 88 - Total da venda: R\$ 1.02	
		Lote: 89 - Total da venda: R\$ 1.02	Lote: 89 - Total da venda: R\$ 1.02	
		Lote: 90 - Total da venda: R\$ 1.02	Lote: 90 - Total da venda: R\$ 1.02	
		Lote: 91 - Total da venda: R\$ 1.02	Lote: 91 - Total da venda: R\$ 1.02	
		Lote: 92 - Total da venda: R\$ 1.02	Lote: 92 - Total da venda: R\$ 1.02	
		Lote: 93 - Total da venda: R\$ 1.02	Lote: 93 - Total da venda: R\$ 1.02	
		Lote: 94 - Total da venda: R\$ 1.02	Lote: 94 - Total da venda: R\$ 1.02	

Input	Expected	Got	
	Lote: 95 - Total da venda: R\$ 1.02 Lote: 96 - Total da venda: R\$ 1.02 Lote: 97 - Total da venda: R\$ 1.02 Lote: 98 - Total da venda: R\$ 1.02 Lote: 99 - Total da venda: R\$ 1.02 Lote: 100 - Total da venda: R\$ 1.02 Lote: 101 - Total da venda: R\$ 1.02 Lote: 102 - Total da venda: R\$ 1.02 Lote: 103 - Total da venda: R\$ 1.02 Lote: 104 - Total da venda: R\$ 1.02 Lote: 105 - Total da venda: R\$ 1.02 Lote: 106 - Total da venda: R\$ 1.02 Lote: 107 - Total da venda: R\$ 1.02 Lote: 108 - Total da venda: R\$ 1.02 Lote: 109 - Total da venda: R\$ 1.02 Lote: 110 - Total da venda: R\$ 1.02 Lote: 111 - Total da venda: R\$ 1.02 Lote: 112 - Total da venda: R\$ 1.02 Lote: 113 - Total da venda: R\$ 1.02 Lote: 114 - Total da venda: R\$ 1.0 ...snip... e: 890 - Total da venda: R\$ 1.02 Lote: 891 - Total da venda: R\$ 1.02 Lote: 892 - Total da venda: R\$ 1.02 Lote: 893 - Total da venda: R\$ 1.02 Lote: 894 - Total da venda: R\$ 1.02 Lote: 895 - Total da venda: R\$ 1.02 Lote: 896 - Total da venda: R\$ 1.02 Lote: 897 - Total da venda: R\$ 1.02 Lote: 898 - Total da venda: R\$ 1.02 Lote: 899 - Total da venda: R\$ 1.02 Lote: 900 - Total da venda: R\$ 1.02 Lote: 901 - Total da venda: R\$ 1.02 Lote: 902 - Total da venda: R\$ 1.02 Lote: 903 - Total da venda: R\$ 1.02 Lote: 904 - Total da venda: R\$ 1.02 Lote: 905 - Total da venda: R\$ 1.02 Lote: 906 - Total da venda: R\$ 1.02 Lote: 907 - Total da venda: R\$ 1.02 Lote: 908 - Total da venda: R\$ 1.02 Lote: 909 - Total da venda: R\$ 1.02 Lote: 910 - Total da venda: R\$ 1.02	Lote: 95 - Total da venda: R\$ 1.02 Lote: 96 - Total da venda: R\$ 1.02 Lote: 97 - Total da venda: R\$ 1.02 Lote: 98 - Total da venda: R\$ 1.02 Lote: 99 - Total da venda: R\$ 1.02 Lote: 100 - Total da venda: R\$ 1.02 Lote: 101 - Total da venda: R\$ 1.02 Lote: 102 - Total da venda: R\$ 1.02 Lote: 103 - Total da venda: R\$ 1.02 Lote: 104 - Total da venda: R\$ 1.02 Lote: 105 - Total da venda: R\$ 1.02 Lote: 106 - Total da venda: R\$ 1.02 Lote: 107 - Total da venda: R\$ 1.02 Lote: 108 - Total da venda: R\$ 1.02 Lote: 109 - Total da venda: R\$ 1.02 Lote: 110 - Total da venda: R\$ 1.02 Lote: 111 - Total da venda: R\$ 1.02 Lote: 112 - Total da venda: R\$ 1.02 Lote: 113 - Total da venda: R\$ 1.02 Lote: 114 - Total da venda: R\$ 1.0 ...snip... e: 890 - Total da venda: R\$ 1.02 Lote: 891 - Total da venda: R\$ 1.02 Lote: 892 - Total da venda: R\$ 1.02 Lote: 893 - Total da venda: R\$ 1.02 Lote: 894 - Total da venda: R\$ 1.02 Lote: 895 - Total da venda: R\$ 1.02 Lote: 896 - Total da venda: R\$ 1.02 Lote: 897 - Total da venda: R\$ 1.02 Lote: 898 - Total da venda: R\$ 1.02 Lote: 899 - Total da venda: R\$ 1.02 Lote: 900 - Total da venda: R\$ 1.02 Lote: 901 - Total da venda: R\$ 1.02 Lote: 902 - Total da venda: R\$ 1.02 Lote: 903 - Total da venda: R\$ 1.02 Lote: 904 - Total da venda: R\$ 1.02 Lote: 905 - Total da venda: R\$ 1.02 Lote: 906 - Total da venda: R\$ 1.02 Lote: 907 - Total da venda: R\$ 1.02 Lote: 908 - Total da venda: R\$ 1.02 Lote: 909 - Total da venda: R\$ 1.02 Lote: 910 - Total da venda: R\$ 1.02	

Aviso!

?

	Input	Expected	Got	
Aviso!		Lote: 911 - Total da venda: R\$ 1.02	Lote: 911 - Total da venda: R\$ 1.02	
		Lote: 912 - Total da venda: R\$ 1.02	Lote: 912 - Total da venda: R\$ 1.02	
		Lote: 913 - Total da venda: R\$ 1.02	Lote: 913 - Total da venda: R\$ 1.02	
		Lote: 914 - Total da venda: R\$ 1.02	Lote: 914 - Total da venda: R\$ 1.02	
		Lote: 915 - Total da venda: R\$ 1.02	Lote: 915 - Total da venda: R\$ 1.02	
		Lote: 916 - Total da venda: R\$ 1.02	Lote: 916 - Total da venda: R\$ 1.02	
		Lote: 917 - Total da venda: R\$ 1.02	Lote: 917 - Total da venda: R\$ 1.02	
		Lote: 918 - Total da venda: R\$ 1.02	Lote: 918 - Total da venda: R\$ 1.02	
		Lote: 919 - Total da venda: R\$ 1.02	Lote: 919 - Total da venda: R\$ 1.02	
		Lote: 920 - Total da venda: R\$ 1.02	Lote: 920 - Total da venda: R\$ 1.02	
		Lote: 921 - Total da venda: R\$ 1.02	Lote: 921 - Total da venda: R\$ 1.02	
		Lote: 922 - Total da venda: R\$ 1.02	Lote: 922 - Total da venda: R\$ 1.02	
		Lote: 923 - Total da venda: R\$ 1.02	Lote: 923 - Total da venda: R\$ 1.02	
		Lote: 924 - Total da venda: R\$ 1.02	Lote: 924 - Total da venda: R\$ 1.02	
		Lote: 925 - Total da venda: R\$ 1.02	Lote: 925 - Total da venda: R\$ 1.02	
		Lote: 926 - Total da venda: R\$ 1.02	Lote: 926 - Total da venda: R\$ 1.02	
		Lote: 927 - Total da venda: R\$ 1.02	Lote: 927 - Total da venda: R\$ 1.02	
		Lote: 928 - Total da venda: R\$ 1.02	Lote: 928 - Total da venda: R\$ 1.02	
		Lote: 929 - Total da venda: R\$ 1.02	Lote: 929 - Total da venda: R\$ 1.02	
		Lote: 930 - Total da venda: R\$ 1.02	Lote: 930 - Total da venda: R\$ 1.02	
		Lote: 931 - Total da venda: R\$ 1.02	Lote: 931 - Total da venda: R\$ 1.02	
		Lote: 932 - Total da venda: R\$ 1.02	Lote: 932 - Total da venda: R\$ 1.02	
		Lote: 933 - Total da venda: R\$ 1.02	Lote: 933 - Total da venda: R\$ 1.02	
		Lote: 934 - Total da venda: R\$ 1.02	Lote: 934 - Total da venda: R\$ 1.02	
		Lote: 935 - Total da venda: R\$ 1.02	Lote: 935 - Total da venda: R\$ 1.02	
		Lote: 936 - Total da venda: R\$ 1.02	Lote: 936 - Total da venda: R\$ 1.02	
		Lote: 937 - Total da venda: R\$ 1.02	Lote: 937 - Total da venda: R\$ 1.02	
		Lote: 938 - Total da venda: R\$ 1.02	Lote: 938 - Total da venda: R\$ 1.02	
		Lote: 939 - Total da venda: R\$ 1.02	Lote: 939 - Total da venda: R\$ 1.02	
		Lote: 940 - Total da venda: R\$ 1.02	Lote: 940 - Total da venda: R\$ 1.02	
		Lote: 941 - Total da venda: R\$ 1.02	Lote: 941 - Total da venda: R\$ 1.02	
		Lote: 942 - Total da venda: R\$ 1.02	Lote: 942 - Total da venda: R\$ 1.02	
		Lote: 943 - Total da venda: R\$ 1.02	Lote: 943 - Total da venda: R\$ 1.02	
		Lote: 944 - Total da venda: R\$ 1.02	Lote: 944 - Total da venda: R\$ 1.02	
		Lote: 945 - Total da venda: R\$ 1.02	Lote: 945 - Total da venda: R\$ 1.02	
		Lote: 946 - Total da venda: R\$ 1.02	Lote: 946 - Total da venda: R\$ 1.02	
		Lote: 947 - Total da venda: R\$ 1.02	Lote: 947 - Total da venda: R\$ 1.02	
		Lote: 948 - Total da venda: R\$ 1.02	Lote: 948 - Total da venda: R\$ 1.02	
		Lote: 949 - Total da venda: R\$ 1.02	Lote: 949 - Total da venda: R\$ 1.02	
		Lote: 950 - Total da venda: R\$ 1.02	Lote: 950 - Total da venda: R\$ 1.02	
		Lote: 951 - Total da venda: R\$ 1.02	Lote: 951 - Total da venda: R\$ 1.02	

	Input	Expected	Got	
Aviso!		Lote: 952 - Total da venda: R\$ 1.02	Lote: 952 - Total da venda: R\$ 1.02	
		Lote: 953 - Total da venda: R\$ 1.02	Lote: 953 - Total da venda: R\$ 1.02	
		Lote: 954 - Total da venda: R\$ 1.02	Lote: 954 - Total da venda: R\$ 1.02	
		Lote: 955 - Total da venda: R\$ 1.02	Lote: 955 - Total da venda: R\$ 1.02	
		Lote: 956 - Total da venda: R\$ 1.02	Lote: 956 - Total da venda: R\$ 1.02	
		Lote: 957 - Total da venda: R\$ 1.02	Lote: 957 - Total da venda: R\$ 1.02	
		Lote: 958 - Total da venda: R\$ 1.02	Lote: 958 - Total da venda: R\$ 1.02	
		Lote: 959 - Total da venda: R\$ 1.02	Lote: 959 - Total da venda: R\$ 1.02	
		Lote: 960 - Total da venda: R\$ 1.02	Lote: 960 - Total da venda: R\$ 1.02	
		Lote: 961 - Total da venda: R\$ 1.02	Lote: 961 - Total da venda: R\$ 1.02	
		Lote: 962 - Total da venda: R\$ 1.02	Lote: 962 - Total da venda: R\$ 1.02	
		Lote: 963 - Total da venda: R\$ 1.02	Lote: 963 - Total da venda: R\$ 1.02	
		Lote: 964 - Total da venda: R\$ 1.02	Lote: 964 - Total da venda: R\$ 1.02	
		Lote: 965 - Total da venda: R\$ 1.02	Lote: 965 - Total da venda: R\$ 1.02	
		Lote: 966 - Total da venda: R\$ 1.02	Lote: 966 - Total da venda: R\$ 1.02	
		Lote: 967 - Total da venda: R\$ 1.02	Lote: 967 - Total da venda: R\$ 1.02	
		Lote: 968 - Total da venda: R\$ 1.02	Lote: 968 - Total da venda: R\$ 1.02	
		Lote: 969 - Total da venda: R\$ 1.02	Lote: 969 - Total da venda: R\$ 1.02	
		Lote: 970 - Total da venda: R\$ 1.02	Lote: 970 - Total da venda: R\$ 1.02	
		Lote: 971 - Total da venda: R\$ 1.02	Lote: 971 - Total da venda: R\$ 1.02	
		Lote: 972 - Total da venda: R\$ 1.02	Lote: 972 - Total da venda: R\$ 1.02	
		Lote: 973 - Total da venda: R\$ 1.02	Lote: 973 - Total da venda: R\$ 1.02	
		Lote: 974 - Total da venda: R\$ 1.02	Lote: 974 - Total da venda: R\$ 1.02	
		Lote: 975 - Total da venda: R\$ 1.02	Lote: 975 - Total da venda: R\$ 1.02	
		Lote: 976 - Total da venda: R\$ 1.02	Lote: 976 - Total da venda: R\$ 1.02	
		Lote: 977 - Total da venda: R\$ 1.02	Lote: 977 - Total da venda: R\$ 1.02	
		Lote: 978 - Total da venda: R\$ 1.02	Lote: 978 - Total da venda: R\$ 1.02	
		Lote: 979 - Total da venda: R\$ 1.02	Lote: 979 - Total da venda: R\$ 1.02	
		Lote: 980 - Total da venda: R\$ 1.02	Lote: 980 - Total da venda: R\$ 1.02	
		Lote: 981 - Total da venda: R\$ 1.02	Lote: 981 - Total da venda: R\$ 1.02	
		Lote: 982 - Total da venda: R\$ 1.02	Lote: 982 - Total da venda: R\$ 1.02	
		Lote: 983 - Total da venda: R\$ 1.02	Lote: 983 - Total da venda: R\$ 1.02	
		Lote: 984 - Total da venda: R\$ 1.02	Lote: 984 - Total da venda: R\$ 1.02	
		Lote: 985 - Total da venda: R\$ 1.02	Lote: 985 - Total da venda: R\$ 1.02	
		Lote: 986 - Total da venda: R\$ 1.02	Lote: 986 - Total da venda: R\$ 1.02	
		Lote: 987 - Total da venda: R\$ 1.02	Lote: 987 - Total da venda: R\$ 1.02	
		Lote: 988 - Total da venda: R\$ 1.02	Lote: 988 - Total da venda: R\$ 1.02	
		Lote: 989 - Total da venda: R\$ 1.02	Lote: 989 - Total da venda: R\$ 1.02	
		Lote: 990 - Total da venda: R\$ 1.02	Lote: 990 - Total da venda: R\$ 1.02	
		Lote: 991 - Total da venda: R\$ 1.02	Lote: 991 - Total da venda: R\$ 1.02	
		Lote: 992 - Total da venda: R\$ 1.02	Lote: 992 - Total da venda: R\$ 1.02	

	Input	Expected	Got	
		Lote: 993 - Total da venda: R\$ 1.02 Lote: 994 - Total da venda: R\$ 1.02 Lote: 995 - Total da venda: R\$ 1.02 Lote: 996 - Total da venda: R\$ 1.02 Lote: 997 - Total da venda: R\$ 1.02 Lote: 998 - Total da venda: R\$ 1.02 Lote: 999 - Total da venda: R\$ 1.02 Lote: 1000 - Total da venda: R\$ 1.02	Lote: 993 - Total da venda: R\$ 1.02 Lote: 994 - Total da venda: R\$ 1.02 Lote: 995 - Total da venda: R\$ 1.02 Lote: 996 - Total da venda: R\$ 1.02 Lote: 997 - Total da venda: R\$ 1.02 Lote: 998 - Total da venda: R\$ 1.02 Lote: 999 - Total da venda: R\$ 1.02 Lote: 1000 - Total da venda: R\$ 1.02	

Passou em todos os teste! ✓

Question author's solution (Python3):

```

1 cotacao = float(input())
2 tam_lote = int(input())
3 lotes = int(input())
4
5 correcao = 1 + 0.025
6 i = 1
7 while i <= lotes:
8     total = (cotacao * tam_lote) * correcao
9     total_formatado = "{:.2f}".format(total)
10    print('Lote: ' + str(i) + ' - Total da venda: R$ ' + str(total_formatado))
11    i += 1
12
13 #Solução alternativa
14 """
15

```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 3

Correto

Atingiu 1,00 de 1,00

Bom dia e boa noite

Crie um [programa](#) que lê um número inteiro positivo n e realiza uma [iteração](#) que repete os seguintes passos:

Se n é zero, imprima "Ué? Já acabou?" e finalize a [iteração](#).

Se n é par, imprima "Bom dia!".

Se n é ímpar, imprima "Boa noite!".

Ao fim de cada [iteração](#), subtraia 1 de n .

Entrada

A entrada contém um número inteiro positivo.

Saída

A saída consiste em uma série de mensagens imprimidas, dependendo da entrada.

Observações

- No primeiro exemplo de teste, foi digitado 1, retornando "Boa noite!" e "Ué? Já acabou?".
- No segundo exemplo de teste, foi digitado 2, retornando "Bom dia!", "Boa noite!" e "Ué? Já acabou?".
- No terceiro exemplo de teste, foi digitado 3, retornando "Boa noite!", "Bom dia!", "Boa noite!" e "Ué? Já acabou?".

For example:

Input	Result
-------	--------

Aviso!

Input	Result
1	Boa noite! Ué? Já acabou?
2	Bom dia! Boa noite! Ué? Já acabou?
3	Boa noite! Bom dia! Boa noite! Ué? Já acabou?

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 n = int(input())
2
3 def mensagem(n):
4     if n == 0:
5         print("Ué? Já acabou?")
6     elif n % 2 == 0:
7         print("Bom dia!")
8     else:
9         print("Boa noite!")
10
11 while n >= 0:
12     mensagem(n)
13     n -= 1

```

	Input	Expected	Got	
--	-------	----------	-----	--

Aviso!

?

	Input	Expected	Got	
✓	1	Boa noite! Ué? Já acabou?	Boa noite! Ué? Já acabou?	✓
✓	2	Bom dia! Boa noite! Ué? Já acabou?	Bom dia! Boa noite! Ué? Já acabou?	✓
✓	3	Boa noite! Bom dia! Boa noite! Ué? Já acabou?	Boa noite! Bom dia! Boa noite! Ué? Já acabou?	✓
✓	5	Boa noite! Bom dia! Boa noite! Bom dia! Boa noite! Ué? Já acabou?	Boa noite! Bom dia! Boa noite! Bom dia! Boa noite! Ué? Já acabou?	✓
✓	4	Bom dia! Boa noite! Bom dia! Boa noite! Ué? Já acabou?	Bom dia! Boa noite! Bom dia! Boa noite! Ué? Já acabou?	✓
✓	10	Bom dia! Boa noite! Bom dia! Boa noite! Bom dia! Boa noite! Bom dia! Boa noite! Bom dia! Boa noite! Ué? Já acabou?	Bom dia! Boa noite! Bom dia! Boa noite! Bom dia! Boa noite! Bom dia! Boa noite! Bom dia! Boa noite! Ué? Já acabou?	✓

Passou em todos os teste! ✓

Aviso!

?

Question author's solution (Python3):

```
1 n = int(input())
2
3 while True:
4     if n == 0:
5         print("Ué? Já acabou?")
6         break
7     if n % 2 == 0:
8         print("Bom dia!")
9     else:
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 4

Correto

Atingiu 1,00 de 1,00

Saudações

Crie um [programa](#) que lê um número inteiro positivo n e que, por meio de uma [iteração](#), chama a [função](#) *saudacoes* n vezes. A [função](#) *saudacoes* imprime a mensagem "Saudações terráqueo! Eu tenho perda de [memória](#) recente...".

Entrada

A entrada contém um número inteiro.

Saída

A saída consiste em n mensagens repetidas.

Observações

- No primeiro exemplo de teste, foi digitado 1, retornando "Saudações terráqueo! Eu tenho perda de [memória](#) recente...".
- No segundo exemplo de teste, foi digitado 2, retornando "Saudações terráqueo! Eu tenho perda de [memória](#) recente..." e "Saudações terráqueo! Eu tenho perda de [memória](#) recente...".
- No terceiro exemplo de teste, foi digitado 3, retornando três repetições da mensagem "Saudações terráqueo! Eu tenho perda de [memória](#) recente...".

Particularidade do Tópico

Atenção, a criação de uma [função](#) com o nome determinado pelo enunciado é fundamental para a prática do aluno e o Moodle irá descontar pontos caso a criação não tenha sido feita corretamente (sendo *case-sensitive* o nome da [função](#)).

For example:

Input	Result
	Saudações terráqueo! Eu tenho perda de memória recente...

Aviso!

Input	Result
2	Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente...
3	Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente...

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 ▼ def saudacoes():
2     print("Saudações terráqueo! Eu tenho perda de memória recente...")
3
4     n = int(input())
5
6 ▼ for x in range(n):
7     saudacoes()

```

	Input	Expected	Got	
✓	1	Saudações terráqueo! Eu tenho perda de memória recente...	Saudações terráqueo! Eu tenho perda de memória recente...	✓

Aviso!

?

	Input	Expected	Got	
✓	2	Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente...	Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente...	✓
✓	3	Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente...	Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente...	✓
✓	5	Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente...	Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente...	✓
✓	8	Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente...	Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente...	✓

Aviso!

?

	Input	Expected	Got	
✓	16	Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente...	Saudações terráqueo! Eu tenho perda de memória recente... Saudações terráqueo! Eu tenho perda de memória recente...	✓

Passou em todos os teste! ✓

Question author's solution (Python3):

Aviso!

```
def saudacoes():
```

```
    print("Saudações terráqueo! Eu tenho perda de memória recente...")
```



```
2 |         print( "Saudações Cerraqueo! Eu tenho perda de memória recente..." )
3 |
4 | n = int(input())
5 | while n != 0:
6 |     ,      ,
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 5

Correto

Atingiu 1,00 de 1,00

Maior e Menor

Codifique um [programa](#) que leia uma quantidade $n \geq 1$ de números e diga qual é o menor e qual é o maior valor.

Entrada

A primeira linha contém um inteiro $n \geq 1$. As próximas n linhas contém inteiros $-10^4 \leq x \leq 10^4$.

Saída

Imprima o menor e maior valores da sequência, como é mostrado nos exemplos.

For example:

Input	Result
4 139 257 34 16	Menor: 16 Maior: 257
8 237 473 589 554 842 794 925 629	Menor: 237 Maior: 925
1 12385	Menor: 12385 Maior: 12385

Aviso!

?

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 qtd = int(input())
2
3 entrada = int(input())
4
5 menor = entrada
6 maior = entrada
7
8 for x in range(qtd-1):
9     entrada = int(input())
10    if entrada > maior:
11        maior = entrada
12    if entrada < menor:
13        menor = entrada
14
15 print(f"Menor: {menor} \nMaior: {maior}")

```

	Input	Expected	Got	
✓	4 139 257 34 16	Menor: 16 Maior: 257	Menor: 16 Maior: 257	✓
✓	8 237 473 589 554 842 794 925 629	Menor: 237 Maior: 925	Menor: 237 Maior: 925	✓

Aviso!

?

	Input	Expected	Got	
✓	1 12385	Menor: 12385 Maior: 12385	Menor: 12385 Maior: 12385	✓
✓	3 652 1598 -1	Menor: -1 Maior: 1598	Menor: -1 Maior: 1598	✓
✓	14 561 216 4562 1874 41 1 984652 -8441 654 284 75 15 354 1236	Menor: -8441 Maior: 984652	Menor: -8441 Maior: 984652	✓
✓	3 -39 9666 -764	Menor: -764 Maior: 9666	Menor: -764 Maior: 9666	✓

Passou em todos os teste! ✓

Question author's solution (Python3):

```

1 a = int(input())
2 menor = 1000000
3 maior = -1000000
4 i = 0
5 while i < a:
6     n = int(input())

```

Aviso!

while i < a:

n = int(input())

?

```
6      n = int(input())
7      if n > maior:
8          maior = n
9      if n < menor:
10         menor = n
11         .
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 6

Correto

Atingiu 1,00 de 1,00

Ajuda necessária

A Associação de Pedreiros de Ceilândia (**APC**) está precisando urgentemente de uma pessoa que compute o número com a maior quantidade de divisores entre 1 e n ! Agora você deve estar se perguntando, por quê? A resposta é óbvia: como todos os pedreiros são perfeccionistas, eles querem saber em quantas partes podem dividir uma parede, ou o chão, então eles querem escolher o número ideal n para que possam usar os tamanhos que quiserem. Gui está concorrendo a presidência de **APC** e uma de suas promessas é de já disponibilizar essa funcionalidade para todos os associados, mas Gui não se dá muito bem com [programação](#). Apesar de vocês não se falarem há mais de 10 anos, você topou, pelo bem da amizade que existiu um dia entre vocês, implementar essa funcionalidade.

Entrada

A entrada consiste em um inteiro n ($1 \leq n \leq 1000$).

Saída

Seu [programa](#) deve imprimir dois números inteiros x e d , correspondendo ao menor número com o maior número de divisores entre 1 e n e quantos divisores ele possui, respectivamente.

Observações

- No segundo caso de teste, o resultado é 43 pois 4 é o maior número com a maior quantidade de divisores no intervalo e 3 é a quantidade de divisores.

For example:

Input	Result
1	1 1
Aviso!	3

Input	Result
10	6 4

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 def getPrimos(num):
2     primos = []
3     for numero in range(2,num+1):
4         if isPrimo(numero):
5             primos.append(numero)
6     return primos
7
8
9 def isPrimo(num):
10    cont = 2
11    while cont**2 <= num:
12        if num % cont == 0:
13            return False
14        cont+=1
15    return True

```

	Input	Expected	Got	
✓	1	1 1	1 1	✓
✓	5	4 3	4 3	✓
✓	10	6 4	6 4	✓
✓	100	60 12	60 12	✓

Aviso!

?

	Input	Expected	Got	
✓	900	840 32	840 32	✓
✓	50	48 10	48 10	✓

Passou em todos os teste! ✓

Para resolver o problema, deve-se percorrer entre 1 e n calculando o número de divisores de cada número e salvando o número com o maior número de divisores. Pode-se usar o crivo de eratóstenes para uma melhor eficiência do programa.

Question author's solution (Python3):

```
1 x = int(input())
2 maior = 1
3 divisores = 1
4
5 for numero in range(1, x+1):
6     div = 0
7     for possivelDivisor in range(1, numero+1):
8         if numero%possivelDivisor == 0:
9             div += 1
10
11     if div > divisores:
12         maior = numero
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 7

Correto

Atingiu 1,00 de 1,00

Pescaria

Crie um [programa](#) que lê um número inteiro positivo n e uma [string](#) *peixe*.

Se *peixe* for "atum", imprima a mensagem "Nossa, pesquei um atum gigante!" n vezes por meio de uma [iteração](#).

Se *peixe* for "salmão", imprima a mensagem "Que salmão bonito que pesquei!" $n * 2$ vezes por meio de uma [iteração](#).

Se não for nenhum dos dois, imprima a mensagem "Nem sabia que esse peixe existia."

Entrada

A entrada contém um número inteiro positivo e uma [string](#).

Saída

A saída consiste em uma série de mensagens impressas dependendo da entrada.

Observações

- No primeiro exemplo de teste, foi digitado 1 e "salmão", retornando "Que salmão bonito que pesquei!" e "Que salmão bonito que pesquei!".
- No segundo exemplo de teste, foi digitado 2 e "atum", retornando "Nossa, pesquei um atum gigante!" e "Nossa, pesquei um atum gigante!".
- No terceiro exemplo de teste, foi digitado 1 e "nemo", retornando "Nem sabia que esse peixe existia".

For example:

Input	Result
Aviso!	

Input	Result
1 salmão	Que salmão bonito que pesquei! Que salmão bonito que pesquei!
2 atum	Nossa, pesquei um atum gigante! Nossa, pesquei um atum gigante!
1 nemo	Nem sabia que esse peixe existia

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 qtd = int(input())
2 peixe = input()
3
4 def mensagem(peixe, qtd):
5     if peixe == "salmão":
6         print(2*qtd*"Que salmão bonito que pesquei!\n")
7     elif peixe == "atum":
8         print(qtd*"Nossa, pesquei um atum gigante!\n")
9     else:
10        print("Nem sabia que esse peixe existia")
11
12 mensagem(peixe, qtd)
13

```

	Input	Expected	Got	
✓	1 salmão	Que salmão bonito que pesquei! Que salmão bonito que pesquei!	Que salmão bonito que pesquei! Que salmão bonito que pesquei!	✓

Aviso!

?

	Input	Expected	Got	
✓	2 atum	Nossa, pesquei um atum gigante! Nossa, pesquei um atum gigante!	Nossa, pesquei um atum gigante! Nossa, pesquei um atum gigante!	✓
✓	1 nemo	Nem sabia que esse peixe existia	Nem sabia que esse peixe existia	✓
✓	1 atum	Nossa, pesquei um atum gigante!	Nossa, pesquei um atum gigante!	✓
✓	4 salmão	Que salmão bonito que pesquei! Que salmão bonito que pesquei! Que salmão bonito que pesquei! Que salmão bonito que pesquei! Que salmão bonito que pesquei! Que salmão bonito que pesquei! Que salmão bonito que pesquei! Que salmão bonito que pesquei!	Que salmão bonito que pesquei! Que salmão bonito que pesquei! Que salmão bonito que pesquei! Que salmão bonito que pesquei! Que salmão bonito que pesquei! Que salmão bonito que pesquei! Que salmão bonito que pesquei! Que salmão bonito que pesquei!	✓
✓	43 linguado	Nem sabia que esse peixe existia	Nem sabia que esse peixe existia	✓

Passou em todos os teste! ✓

Question author's solution (Python3):

```

1 | n = int(input())
2 | peixe = input()
3 |
4 | if peixe == "salmão":
5 |     n *= 2
6 |     while n != 0:
7 |         print("Que salmão bonito que pesquei!")
8 |         n -= 1
9 | elif peixe == "atum":
10 |    while n != 0:
        print("Nossa, pesquei um atum gigante!")
        n -= 1

```

Aviso!

?

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 8

Correto

Atingiu 1,00 de 1,00

ICPC

Um dia, os três amigos Leakim, Sarev e Odranoel decidiram fazer um time e participar de competições de [programação](#). Os participantes geralmente recebem vários problemas durante essas competições. Bem antes do início, os amigos decidiram que só iriam implementar a solução de um problema se pelo menos dois deles tivessem certeza da solução. Caso contrário, eles não iriam escrever a solução.

Dessa forma, [dado](#) que a competição oferece n problemas aos participantes, e sabendo quais amigos tem certeza das soluções, ajude-os a encontrar o número de problemas para os quais eles vão escrever uma solução.

Entrada

A primeira linha de entrada contém um único inteiro $1 \leq n \leq 1000$ referente ao número de problemas na competição. Depois, n linhas com três inteiros cada, sendo cada inteiro 0 ou 1. Se o primeiro número da linha é igual a 1, então Leakim tem certeza da solução, caso contrário ele não sabe. O segundo número mostra a visão de Sarev da solução e o terceiro número mostra a visão de Odranoel. Os números nas linhas são separados por espaços.

Saída

Imprima um único inteiro indicando o número de problemas que os amigos vão implementar na competição.

For example:

Input	Result
3 1 1 0 1 1 1 1 0 0	2
2 1 0 0 0 1 1	1

Aviso!

?

Input	Result
1 1 0 0	0

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 total_problemas = int(input())
2
3 qtd_certezas = 0
4
5 for n in range(total_problemas):
6     temp = 0
7     certasas_n = input().split(' ')
8     for i in range(3):
9         temp += int(certasas_n[i])
10    if temp >= 2:
11        qtd_certezas += 1
12
13 print(qtd_certezas)

```

	Input	Expected	Got	
✓	3 1 1 0 1 1 1 1 0 0	2	2	✓
✓	2 1 0 0 0 1 1	1	1	✓

Aviso!

?

	Input	Expected	Got	
✓	1 1 0 0	0	0	✓
✓	5 1 0 0 0 1 0 1 1 1 0 0 1 0 0 0	1	1	✓
✓	10 0 1 0 0 1 0 1 1 0 1 0 0 0 0 1 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0	4	4	✓
✓	50 0 0 0 0 1 1 1 1 1 0 1 0 1 0 1 1 1 1 0 0 1 1 0 0 1 1 0 1 0 1 0 1 0 0 0 1 1 1 0 0 1 0 1 1 0 0 0 0 1 1 0 1	29	29	✓

Aviso!

?

	Input	Expected	Got	
	0 0 1			
	1 1 0			
	1 1 1			
	0 1 1			
	1 1 0			
	0 0 0			
	0 0 0			
	1 1 1			
	0 0 0			
	1 1 1			
	0 1 1			
	0 0 1			
	0 0 0			
	0 0 0			
	1 1 0			
	1 1 0			
	1 0 1			
	1 0 0			
	1 0 1			
	1 0 1			
	0 1 1			
	1 1 0			
	1 1 0			
	0 1 0			
	1 0 1			
	0 0 0			
	0 0 0			
	0 0 0			
	0 0 1			
	1 1 1			
	0 1 1			
	1 0 1			

Passou em todos os teste! ✓

Question author's solution (Python3):

Aviso! = int(input())


```
3 ans = 0
4 for i in range(n):
5     a, b, c = [int(x) for x in input().split()]
6     s = a + b + c - 1
7     ans += max(0, min(1, s))
8
9 print(ans)
10
11 #solução alternativa
12 """
13 n = int(input())
14 cont = 0
15 for i in range(1, n+1):
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 9

Correto

Atingiu 1,00 de 1,00

Cerveja Pong



Ontem mesmo o prof Faina encontrou com o prof Emidio (de forma virtual) e tiveram uma conversa legal sobre a Cerveja Pong. Daí ele se perguntou, será que o ano em que o prof Emidio nasceu é primo?

Entrada

A única linha de entrada contém um inteiro $1 \leq n \leq 10^5$ que representa o ano em que Emidio nasceu. Não é necessário validar se os números estão dentro do intervalo definido.

Saída

Imprima "Emidio" se o ano em que Emidio nasceu é primo e "Faina" caso contrário.

Observações

- No primeiro caso de teste, o resultado é "Emídio" pois o número 2 é um número primo.
- No segundo caso de teste, o resultado é "Emídio" pois o número 3 é um número primo.
- No terceiro caso de teste, o resultado é "Faina" pois o número 4 não é um número primo.

For example:

Aviso! **result**

?

Input	Result
2	Emidio
3	Emidio
4	Faina

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 def isPrimo(num):
2     if num >= 2:
3         cont = 2
4         while cont**2 <= num:
5             if num % cont == 0:
6                 return False
7             cont+=1
8         return True
9     else:
10        return False
11
12
13 ano = int(input())
14 if isPrimo(ano):
15     print("Emidio")

```

	Input	Expected	Got	
✓	2	Emidio	Emidio	✓
✓	3	Emidio	Emidio	✓
✓	4	Faina	Faina	✓

Aviso!

?

	Input	Expected	Got	
✓	4531	Faina	Faina	✓
✓	8941	Emidio	Emidio	✓
✓	2049	Faina	Faina	✓

Passou em todos os teste! ✓

Para resolver a questão basta verificar por quais números o valor de entrada é divisível.

Obs.: O número 1 não é um número primo.

Question author's solution (Python3):

```
1 x = int(input())
2
3 i = 2
4 prime = True
5 while i*i <= x and prime:
6     if x % i == 0:
7         prime = False
8     i += 1
9
10 if prime and x > 1:
11     print("Emidio")
```

Correto

Notas para este envio: 1,00/1,00.

Aviso!

?

Questão 10

Correto

Atingiu 1,00 de 1,00

Goldão!

Marcão acabou de entrar na UnB e está cursando APC também. Ele tem um caso muito especial, tem muita facilidade com matemática mas não se dá muito bem com [programação](#). Então se você estiver precisando de ajuda em FTC, C1, IAL, certamente pode ir falar com ele que ele irá te ajudar. Mas como Caetano, um dos professores de CiC, sempre diz "Nenhum almoço é de graça!", então Marcão sempre pede ajuda para você em alguma matéria de computação. E agora é seu momento de brilhar! Marcão aprendeu uma nova conjectura, a Conjectura de Goldbach (clique [aqui](#) para saber mais). Ele conseguiu entender, mas queria calcular para um [dado](#) n par, quais seriam os dois primos a, b tais que $a + b = n$ e a diferença entre a e b seja a maior possível. Como ele acha cansativo ter que calcular esses números, ele pediu para que você fizesse isso para ele. Assim ele poderia lhe ajudar em outra matéria. Você, frustrado com limites, [vetores](#) e tentar provar que 2 é par, não demorou para aceitar. Você deve então implementar a [função](#) `goldao` que recebe um inteiro n como parâmetro e retorna dois inteiros a, b primos, tais que a soma deles seja n .

Entrada

A entrada consiste no parâmetro da [função](#) `goldao`, que é um número inteiro n ($4 \leq n \leq 10^8$).

Saída

Sua [função](#) deve retornar dois inteiros a, b ($a \leq b$) primos que satisfaçam o enunciado do problema.

Observações

- No primeiro exemplo de teste, o resultado é (2, 2) pois são os dois números primos que somados resultam em 4.
- No segundo exemplo de teste, os primos 3 e 97 somados resultam no valor $n = 100$.

For example:

Test	Result
Aviso!	

Test	Result
print(goldao(4))	(2, 2)
print(goldao(100))	(3, 97)
print(goldao(36))	(5, 31)

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```

1 def isPrimo(num):
2     if num >= 2:
3         cont = 2
4         while cont**2 <= num:
5             if num % cont == 0:
6                 return False
7             cont+=1
8         return True
9     else:
10        return False
11
12
13 def goldao(num):
14     for b in range(num+1):
15         if isPrimo(num) and isPrimo(b):

```

	Test	Expected	Got	
✓	print(goldao(4))	(2, 2)	(2, 2)	✓
✓	print(goldao(100))	(3, 97)	(3, 97)	✓
✓	print(goldao(36))	(5, 31)	(5, 31)	✓

Aviso!

?

	Test	Expected	Got	
✓	print(goldao(1000))	(3, 997)	(3, 997)	✓
✓	print(goldao(556))	(47, 509)	(47, 509)	✓
✓	print(goldao(50836))	(3, 50833)	(3, 50833)	✓

Passou em todos os teste! ✓

Conforme o enunciado, queremos determinar os dois números primos (a) e (b) que, somados, resultam no valor (n) da entrada. Levando em conta que $(n=a+b)$, podemos elaborar uma [estrutura de repetição](#) variando-se o valor de $(a=2, \dots, n-1)$ (lembre-se que o número (1) não é primo), e para cada valor de (a) , deve-se verificar se (a) é primo e se $(b=n-a)$ também é primo. Em caso afirmativo, retornar os dois números.

Question author's solution (Python3):

```
1 def ehPrimo(n):
2
3     if n < 2:
4         return False
5
6     for i in range(2,n):
7         if n % i == 0:
8             return False
9     return True
10
11 def goldao(n):
12
13     for a in range(2,n+1):
14         b = n-a
15         if ehPrimo(a) and ehPrimo(b):
```

Aviso!

este envio: 1,00/1,00.

?



Aviso!

?