

FUNÇÕES

Problema A

Juaozin implementou o seguinte código em C para receber dois números inteiros positivos, X e Y, e calcular a soma de todos no intervalo [X,Y].

```
int main(){
    int x,y;
    scanf("%d %d", &x, &y);
    int resultado = soma(x,y);
    printf("%d\n", resultado);
    return 0;
}
```

Porém ele está com dificuldade de implementar a função soma. Como grande amigo de Juaozin, ele te pediu para implementar a função para ele com a seguinte assinatura.

```
int soma(int x, int y);
```

A função recebe dois inteiros, x e y, e retorna um inteiro que representa a soma de todos os números inteiros no intervalo [X,Y].

Entrada

A entrada é composta por dois inteiros, x e y, separados por espaço. Você pode considerar que X é sempre menor que Y.

Saída

O resultado da soma de todos os números inteiros no intervalo [X,Y].

Exemplo de entrada	Exemplo de saída
1 4	10

Exemplo de entrada	Exemplo de saída
4 6	15

Instruções: Você deve entregar o código com a main que Juozin escreveu!

Problema B

Juaozin precisa de sua ajuda novamente. Ele implementou uma função para verificar se um número é primo ou não.

```
int eh_primo(int x){
    int num_divisores = 0;
    for(int i=1; i<=x; i++){
        if(x%i == 0){
            num_divisores++;
        }
    }
    if(num_divisores == 2){
        return 1;
    }
    return 0;
}
```

Entretanto, Juaozin agora não sabe como implementar uma *main* que recebe um número inteiro e imprime todos os números primos que dividem este número. Dessa forma, como grande amigo, ele pediu sua ajuda novamente para implementar uma *main* que usa a função que ele escreveu para fazer esse cálculo.

Entrada

A entrada é composta por um número inteiro N .

Saída

Todos os números primos que dividem N , um por linha.

Exemplo de entrada	Exemplo de saída
9	3

Exemplo de entrada	Exemplo de saída
15	3 5

Exemplo de entrada	Exemplo de saída
13	13

Exemplo de entrada	Exemplo de saída
30	2 3 5

Instruções: Você deve entregar o código com a função que Juaozin escreveu!

Problema C

Juaozin implementou o seguinte código em C para ler uma frase do terminal e transformar todas as letras do alfabeto da frase lida em minúsculas.

```
int main(){
    char l;
    do{
        scanf("%c", &l);
        if(l == '\n'){
            break;
        }
        printf("%c", minuscula(l));
    } while(1);
    return 0;
}
```

Porém ele está com dificuldade de implementar a função *minuscula*. Pela segunda vez, ele te pede ajuda para implementar esta função para ele. A função deve converter letras do alfabeto de maiúsculas [A-Z] para minúsculas [a-z]. Se a letra já for minúscula, a função deve apenas retornar a letra, sem nenhuma alteração. O mesmo deve acontecer se o caractere não for uma letra do alfabeto.

Porém, desta vez, ele esqueceu de te dar a assinatura da função. Porém, você pode inferi-la pelo código da *main* que ele te forneceu.

Entrada

A entrada é composta por uma frase.

Saída

A frase com todos os caracteres minúsculos, seguido de uma qu

Exemplo de entrada	Exemplo de saída
Essa Vale 2 Pontos!	essa vale 2 pontos!

Exemplo de entrada	Exemplo de saída
JUAOZIN, o Amigao.	juaozin, o amigao.

Instruções: Você deve entregar o código com a main que Juaozin escreveu!

Tabela ASC II

Dec	Char	Dec	Char	Dec	Char
32	Space	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	\$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(72	H	104	h
41)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[123	{
60	<	92	\	124	
61	=	93]	125	}
62	>	94	^	126	~
63	?	95	_	127	DEL

Problema D

Faça uma função com protótipo `int somaAlgarismos(int x)`; que recebe um número inteiro maior do que zero e retorna a soma de todos os seus algarismos. Por exemplo, `somaAlgarismos(251)` corresponderá ao valor 8 ($2 + 5 + 1$). Se o número lido não for maior ou igual a zero, a função deverá retornar -1.

Em seguida, escreva um programa completo (*main*) que use essa função para realizar a soma de um número inserido pelo usuário.

Entrada

A entrada é composta por um inteiro N ($-10^9 < N < 10^9$).

Saída

A soma dos algarismos de N , caso ele seja positivo ou “Negativo” caso N seja menor que 0.

Exemplo de entrada	Exemplo de saída
1453	13

Exemplo de entrada	Exemplo de saída
-4	Negativo

Instruções: Você deve entregar o código com a main que Juaozin escreveu!

Problema E

Faça um procedimento em C que receba como parâmetros um valor inteiro N e uma letra L . O procedimento deve imprimir um triângulo invertido de N linhas com a letra L , conforme o exemplo abaixo (para $N = 5$ e $L = 'k'$).

```
kkkkk
kkkk
kkk
kk
k
```

Em seguida, escreva um programa completo (*main*) que leia o número N e o caractere L do usuário e use esse procedimento para imprimir o padrão.

Entrada

A entrada é composta por um inteiro N , um espaço em branco e o caractere L . Você pode considerar que N sempre será maior que 0 e que L sempre será uma letra [a,z].

Saída

O padrão do enunciado.

Exemplo de entrada	Exemplo de saída
3 x	xxx xx x

Exemplo de entrada	Exemplo de saída
6 b	bbbbbb bbbbb bbbb bbb bb b

Problema F

Juaozin está desenvolvendo um jogo bi-dimensional onde o personagem principal pode atirar nos inimigos. Ele gostaria que você desenvolvesse uma função com o seguinte protótipo.

```
int retaPassaPeloPonto(float m, float b, float x1, float y1);
```

A função deve verificar se a reta $y = mx + b$ passa pelo ponto $(x1, y1)$. Em caso positivo, a função deve retornar 1. Em caso negativo, a função deve retornar 0.

Em seguida, faça um programa completo (*main*) que receba do usuário os valores de m , b , $x1$ e $y1$ e use a função desenvolvida para imprimir “Acertou” caso a reta passe pelo ponto ou “Errou”, caso contrário.

Entrada

A entrada é composta por quatro números decimais (*float*), separados por espaço, representando respectivamente m , b , $x1$ e $y1$.

Saída

“Acertou” caso a reta $y = mx + b$ passe pelo ponto $(x1, y1)$ ou “Errou”, caso contrário.

Exemplo de entrada	Exemplo de saída
2.0 4.0 3.0 10.0	Acertou

Exemplo de entrada	Exemplo de saída
-1.0 2.0 3.0 -1.0	Acertou

Exemplo de entrada	Exemplo de saída
2.0 4.0 3.0 9.0	Errou

STRUCTS (Tipos estruturados)

Problema A

Mariazinha encontrou o seguinte trecho de código em C na Internet para ler a informação de dois carros e imprimir o preço e a categoria do carro com ano mais recente.

```
int main(){
    struct carro c1, c2;
    scanf("%lf %d %c", &c1.preco, &c1.ano,
&c1.categoria);    scanf("%lf %d %c", &c2.preco, &c2.ano,
&c2.categoria);
    ...
    return 0;
}
```

Porém Mariazinha não consegue compilar este código pois falta a definição da estrutura *carro* bem como a lógica para imprimir as informações do carro com ano mais recente. Você pode assumir que o ano dos carros serão diferentes.

Entrada

A entrada é composta por duas linhas, cada uma com informações sobre um carro. Cada linha possui um decimal (double), um int e um char, separados por espaço, que representam, respectivamente, o preço, ano e categoria de um carro.

Saída

O preço, com duas casas decimais, e a categoria do carro, separados por espaço.

Exemplo de entrada	Exemplo de saída
25500.00 2010 P 39999.00 2009 S	25500.00 P

Instruções: Você deve entregar o código estendendo o código que Mariazinha encontrou

Problema B

Joãozinho havia planejado rever um filme no final de semana, mas Joãozinho não consegue mais lembrar o ID do filme que ele deseja assistir, por outro lado, ele se lembra bem do ano e da duração desse filme.

Para ajudar Joãozinho, escreva um programa que seja capaz de imprimir o ID do filme que Joãozinho precisa. É garantido que nenhum outro filme possui o mesmo ano e duração que o filme que Joãozinho deseja assistir.

Seu código deve utilizar a seguinte struct Filme:

```
struct Filme{  
    int id;  
    int duracao;  
    int ano;  
};
```

Entrada

A primeira linha da entrada consiste por um inteiro n , a quantidade de filmes. As próximas n linhas possuem id, duração e ano de cada filme, nessa ordem. A última linha da entrada possui a duração e o ano do filme que Joãozinho deseja assistir.

Saída

O ID do filme que Joãozinho precisa.

Exemplo de entrada	Exemplo de saída
4 1001 120 2000 1002 90 2005 1003 60 2003 1004 120 2010 120 2000	1001

Problema C

Uma loja deseja uma forma de encontrar o nome de um cliente, a partir de seu cpf. Cada cliente tem a seguinte estrutura:

```
typedef struct cliente{
    char cpf[20];
    char nome[50];
} Cliente;
```

Sua tarefa é implementar uma função que recebe um vetor de clientes, um cpf, e o tamanho do vetor de clientes, e caso ache um cliente com o cpf igual ao recebido, imprima seu nome. A função deve ter o seguinte protótipo:

```
void procuraCliente(Cliente cliente[], char cpf[], int tamanho);
```

A loja já criou a main, porém de forma parcial que você deve completar:

```
int main(){
    Cliente clientes[10];
    char cpf[20];
    fgets(cpf, 20, stdin);
    for(int i = 0; i < 10; i++){
        fgets(_____, 20, stdin);
        fgets(_____, 50, stdin);
    }
    procuraCliente(clientes, cpf, 10);
    return 0;
}
```

Entrada

A primeira linha é composta pelo cpf a ser buscado. As seguintes linhas alternam entre o cpf e nome dos clientes.

Saída

Seu programa deverá imprimir o nome do cliente com o cpf lido na primeira linha.

Exemplo de entrada	Exemplo de saída
201398877 123456789 Maria Souza 127890100 Ruan Almeida 405431232 Joao Luiz 209543200 Julia Costa 100452491 Beatriz Lima 871923012 Pedro Silva 492738614 Laura Rocha 615384792 Lucas Oliveira 549284674 Ana Santos 201398877 Arthur Morgan	Arthur Morgan

Problema D

O seu chefe solicitou a você para implementar o seguinte programa em C nos passos que seguem:

- Crie uma estrutura em C com nome *struct livro*. O livro deve ter um ISBN (inteiro), o ano de publicação (inteiro) e o preço sugerido (double). Você deve criar o sinônimo *Livro* para a estrutura.
- Em seguida, escreva uma função em C com a seguinte assinatura.

```
Livro retornaLivroMaisCaro(Livro l1, Livro l2);
```

que recebe dois livros e retorna o livro com maior preço sugerido. Em caso de empate, retornar o l1.

- Por fim, faça um programa completo que use a função. O seu programa deve criar duas variáveis do tipo Livro e os membros dessas variáveis devem ser preenchidos pelo usuário via teclado. O seu programa então deve chamar a função e imprimir todas as informações do livro com maior preço sugerido.

Uma vez que seu código estiver implementado, ele enviará para a equipe de testes para verificar se está tudo correto com a sua implementação.

Entrada

A entrada é composta por duas linhas, cada uma com informações sobre um livro. Cada linha possui dois inteiros, representando o ISBN e o ano de publicação do livro e, um decimal (float) representando o preço.

Saída

As informações do livro mais caro, separadas por espaço. O preço deve ter precisão de duas casas decimais.

Exemplo de entrada	Exemplo de saída
--------------------	------------------

1111111 2001 15.00 222222 2000 20.00	222222 2000 20.00
---	-------------------

Exemplo de entrada	Exemplo de saída
1111111 2001 20.00 222222 2000 20.00	1111111 2001 20.00

Instruções: Você deve entregar o código estendendo o código que o programador sumido deixou.

Problema E

Mariazinha recebeu agora uma tarefa um pouco mais complexa para casa para criar um programa para simular um combate em um jogo.

```
struct personagem{
    int ataque;
    int defesa;
    int hp;
};

struct jogo{
    struct personagem heroi;
    struct personagem inimigo;
};

int main(){
    struct jogo j;
    // ataque, defesa, hp do heroi
    scanf("%d %d %d", _____, _____, _____);
    // ataque, defesa, hp do inimigo
    scanf("%d %d %d", _____, _____, _____);
    if(simulaCombate(j) == 1){
        printf("Herói ganhou\n");
    } else {
        printf("Inimigo ganhou\n");
    }
    return 0;
}
```

O combate do jogo é realizado em turnos. Todo turno, o heroi e o inimigo possuem um ataque cada e o heroi sempre ataca primeiro. O cálculo do ataque é simples: subtrai a defesa do defensor do ataque do atacante e o resultado é diminuído dos pontos de vida (hp) do defensor. Por exemplo, suponha que o heroi tenha ataque=10, defesa=5 e hp=30 enquanto o inimigo tem ataque=5, defesa=7 e hp=5. Se o heroi realizar um ataque no inimigo, o resultado será 3 (10-7), significando que o inimigo perderá 3 pontos de vida, ou seja, o hp do inimigo reduzirá a 2.

Sua tarefa é implementar a função *simulaCombate* e ajudar Mariazinha a fazer o código do jogo funcionar corretamente.

Você pode considerar que o inimigo e o heroi sempre causarão dano!

Entrada

A entrada é composta pelas informações do heroi e dos inimigos

Saída

Já especificada no problema.

Exemplo de entrada	Exemplo de saída
10 5 10 6 5 10	Heroi ganhou

Exemplo de entrada	Exemplo de saída
10 4 10 5 9 10	Heroi ganhou

Exemplo de entrada	Exemplo de saída
10 2 10 5 8 10	Inimigo ganhou

Exemplo de entrada	Exemplo de saída
10 4 10 5 9 11	Inimigo ganhou

Instruções: Você deve entregar o código com a main que a Mariazinha forneceu.

Para arredondar um número decimal para cima, você pode usar a função `ceil()` da `math.h`, que recebe um número decimal e arredonda para o próximo número inteiro.

```
#include <stdio.h>
#include <math.h>

int main()
{
    double num = 8.33;
    int result;

    result = ceil(num);
    printf("%.2f = %d", num, result);

    return 0;
}
```