

Oficina de Programação em C (ICP037)

Prof. Ronald Souza – IC/UFRJ

Primeira Prova – 16/05/2024

Questão 1) (1.0 ponto)

Deseja-se implementar uma função que, para 4 inteiros de entrada, correspondentes a início (i), fim (f), posição atual (pa) e deslocamento (d), retorne uma posição nova $pn = pa + d$, **de forma circular**.

Por exemplo:

ENTRADA				SAÍDA
i	f	pa	d	pn
90	200	191	7	198
90	200	192	7	199
90	200	193	7	200
90	200	194	7	90
90	200	195	7	91

Assume-se que os valores de entrada são válidos, ou seja, que já estão defendidos externamente. Portanto, *é garantido que $i \leq f$, $i \leq pa \leq f$, e $d \geq 0$* . Foi iniciada a escrita da função, conforme o código abaixo:

```
int pos(int i, int f, int pa, int d) {  
    // 0 que retornar?  
}
```

→ No seu caderno de respostas (**e não aqui!**), indique claramente, dentre as alternativas abaixo, a que contém o corpo da função que captura corretamente o comportamento acima descrito. Para isso, **escreva a letra escolhida e também todo o seu conteúdo**:

- A) `return ((pa + d - f) % (f - i + 1)) + f;`
- B) `return ((pa + d + i) % (f - i + 1)) - i;`
- C) `return ((pa + d - i) / (f - i + 1)) + i;`
- D) `return ((pa + d - i) % (f - i + 1)) + i;`
- E) `return ((pa + d + f) % (f - i + 1)) - f;`

Questão 2) (2.5 pontos)

Escreva um **programa** em C que leia do teclado 2 inteiros a e b **não negativos** e imprima "verdadeiro" caso a soma dos dígitos de a seja igual à soma dos dígitos de b , ou imprima "falso", caso contrário. Seu programa deve necessariamente conter as duas funções a seguir:

- **main**: onde os inteiros são lidos do teclado e o resultado final é impresso.
- **somaDig**: função que recebe um **único** inteiro de entrada e retorna a soma de seus dígitos.

Por exemplo, as entradas abaixo produzirão as seguintes saídas:

ENTRADA		SAÍDA
a	b	
1021	31	"verdadeiro" (pois $1+0+2+1 = 3+1 = 4$)
684	9009	"verdadeiro" (pois $6+8+4 = 9+0+0+9 = 18$)
58	44	"falso" (pois $5+8 = 13$, mas $4+4 = 8$)

Questão 3) (1.5 pontos)

Escreva uma **função** que receba dois inteiros x e y de entrada. Assuma que x e y **já estão defendidos**, de tal forma que, garantidamente, $x \geq 0$, $y \geq 0$ e $x < y$. Sua função deverá **imprimir em ordem crescente** todos os inteiros que são múltiplos, **ao mesmo tempo**, de 2 e de 3, dentro do intervalo fechado $[x, y]$ (i.e. incluindo-se x e y , **se for o caso**). Sua função **não deve retornar valores**.

Questão 4) (2.5 pontos)

Embora o ouvido humano perceba frequências sonoras que vão desde 20 Hertz (Hz) até 20000 Hz, sua sensibilidade é maior para frequências intermediárias, entre 2000 e 5000 Hz.

Ou seja, para um mesmo volume de reprodução, sons na faixa entre 2000 e 5000 Hz são *mais audíveis* por seres humanos, em comparação às demais frequências do espectro auditivo. Assumindo-se valores de entrada **já defendidos**, tais que, garantidamente, são todos ≥ 0 , *sem limite superior*, execute as seguintes tarefas:

- (1.0 ponto) Escreva uma **função** que receba 3 **racionais** de entrada, correspondentes a frequências sonoras quaisquer f_1 , f_2 , e f_3 , e retorne o **inteiro**
1, caso **ao menos uma** dentre as frequências de entrada pertença à faixa mais audível;
0, caso contrário.
- (1.5 pontos) Assuma que a frequência sonora mais sensível ao ouvido humano seja 3500 Hz. Escreva uma **função** que receba 3 **racionais** de entrada, correspondentes a frequências sonoras quaisquer f_1 , f_2 , e f_3 , e retorne **a mais audível dentre elas**, ou seja, **a frequência que está mais próxima de 3500Hz**.

Questão 5) (2.5 pontos) O que será impresso ao final de cada iteração do programa abaixo? No seu caderno de respostas, faça uma tabela análoga a descrita a seguir, e a preencha com os valores esperados:

```
#include <stdio.h>
int main() {
    float a = 1, b = 1;
    int x = 1;
    for (int i = 1; i <= 3; i++) {
        b += i + i / 2;
        a = i + b / 2;
        x = b - a;
        if (x && i) {
            printf("a = %.1f; b = %.1f; x = %d\n", a, b, x);
            continue;
            ++b;
        }
        a = x / 2;
        printf("a = %.1f; b = %.1f; x = %d\n", a, b, x);
    }
    return 0;
}
```

i	a	b	x
1	?	?	?
2	?	?	?
3	?	?	?