

Questão 1 (valor: 2,5 pontos)

Simule a execução do programa abaixo, usando obrigatoriamente o seu número USP como entrada, destacando a saída do programa no quadro da página a seguir (a saída do programa é tudo que result dos comandos printf). Para efeito de correção só será considerada a saída do programa.

```
#include<stdio.h>
int main(){
    int nusp, w, x, y, z, a, t, i;

    printf("Entre com seu No. USP: ");
    scanf("%d", &nusp);

    w = (nusp/1000) % 10;
    x = (nusp/100) % 10;
    y = (nusp/10) % 10;
    z = nusp % 10;
    i = 0;
    printf("%d: w = %d, x = %d, y = %d, z = %d \n", i, w, x, y, z);

    w = 100 * w;
    y = 10 * y;
    z = z - 10;
    i = i+1;
    printf("%d: w = %d, x = %d, y = %d, z = %d \n", i, w, x, y, z);

    t = 0;
    while (t == 0) {
        t = 1;
        if (w > x) {
            a = w;
            w = x;
            x = a;
            t = 0;
        }
        if (x > y) {
            a = x;
            x = y;
            y = a;
            t = 0;
        }
        if (y > z) {
            a = y;
            y = z;
            z = a;
            t = 0;
        }
        i = i+1;
        printf("%d: w = %d, x = %d, y = %d, z = %d \n", i, w, x, y, z);
    }
    return 0;
}
```

Para efeito de correção só será considerada a saída do programa. A tabela a seguir pode ser usada como rascunho para auxiliar na simulação.

1º Roda →

2º Roda →

3º Roda →

Tempo	W	X	Y	Z	U	V	Q
9898249	8	3	4	9	0	11	9
930398	800	3	40	-1	1	11	11
//	3	40	-1	800	2	0	800
//	3	-1	40	800	3	0	40
//	-1	3	40	800	4	0	3
//	-1	3	40	800	5	1	3

saída

> Entree com casu No. USP: 9898249

> 0: W=8, X=3, Y=4, Z=9

> 1: W=800, X=3, Y=40, Z=-1

> 2: W=3, X=40, Y=-1, Z=800

> 3: W=3, X=-1, Y=40, Z=800

> 4: W=-1, X=3, Y=40, Z=800

> 5: W=-1, X=3, Y=40, Z=800

✓
2.5

Questão 2 (valor: 2,5 pontos)

No EP1, você fez um programa que aprende a prever se o usuário irá escolher um número entre 0 e 1. Faça um programa que lê um número binário, sequência de 0's e 1's, que corresponde a uma jogada de jogadas do usuário, e imprime as seguintes contagens:

- (1) o número de vezes que o jogador humano escolheu 0 dado que na jogada anterior sua escolha foi 0 (variável $\text{comportamento}_{00}$);
- (2) o número de vezes que o jogador humano escolheu 0 dado que na jogada anterior sua escolha foi 1 (variável $\text{comportamento}_{10}$);
- (3) o número de vezes que o jogador humano escolheu 1 dado que na jogada anterior sua escolha foi 0 (variável $\text{comportamento}_{01}$); e
- (4) o número de vezes que o jogador humano escolheu 1 dado que na jogada anterior sua escolha foi 1 (variável $\text{comportamento}_{11}$).

Exemplo:

Sequência de jogadas do usuário: 101100101001010
Contagens de comportamentos do usuário:

$\text{comportamento}_{00} = 2$
 $\text{comportamento}_{10} = 6$
 $\text{comportamento}_{01} = 5$
 $\text{comportamento}_{11} = 1$

`while (true) {`

`n();`

`jogadas, ult-jogadas, contador, partidas`

`comp-00, comp-10, comp-01, comp-11;`

`// com a quantidade de partidas;`

`// & partidas);`

`};`

`<partidas) {`

`// com a jogada";`

`// jogada;`

Questão 3 (valor: 2,5 pontos)

Dados dois inteiros a e b , com $a \geq 0$ e $b \geq 1$, os seguintes trechos de programa têm como objetivo calcular o quociente e o resto da divisão de a por b (sem o uso dos operadores $"/"$ e $"\%$). Na final, a variável q deve ter o valor do quociente (que corresponde a a/b em C) e a variável r deve ter o valor do resto (que corresponde a $a \% b$). Sabendo que os dois trechos de programa dados a seguir estão incorretos, você deve:

1. para cada programa, simular o código para valores de a e b que produzam uma saída incorreta. Você deve escolher esses valores e indicá-los no início da simulação.
2. corrigir cada um dos programas, modificando-os minimamente a fim de eliminar os erros. Você não pode escrever um código novo, devendo manter a estrutura lógica de cada uma das partes, intervindo apenas pontualmente nas instruções que geram os erros. Evidentemente você não pode usar os operadores $"/"$ e $"\%"$.

/* Programa 1 */

```
q = 1;
r = a - b;
while ( r >= b ) {
    r = r - b;
    q = q + 1;
}
```

/* Programa 2 */

```
q = 0;
while ( (q+1)*b <= a ) {
    r = a - q * b;
    q = q + 1;
}
```

Simulação (base em $a < b$)

a	b	Exatidão	q	r	a	b
2	3	↔	1	-2	1	3

0.6

Simulação (base em $a < b$)

a	b	Exatidão	q	r	a	b
2	3	↔	0	2	2	3

0.6

Questão 4 (valor: 2,5 pontos)
 Escreva um programa que lê um número real x (ângulo em radianos) e calcula o arco tangente de x através da série:

2,0

Incluindo todos os termos até que $|(-1)^k \frac{x^{2k+1}}{2k+1}| < 0.00001$ (esse último termo não deve ser incluído).

$$\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots$$

Arquivo: arctan.c

```
#include <stdio.h>
int main() {
    int k;
    float x, termo, arctanx;
    printf("Entre com um valor x: ");
    scanf("%f", &x);
    arctanx = 0;
    termo = x;
    k = 1;
    while (termo > 0.00001 || termo < -0.00001) {
        arctanx = arctanx + termo;
        termo = ((-1)*termo * x*x)/(k+2);
        k = k+2;
    }
    printf("Arco tangente de %d eh %d", x, arctanx);
}
```

↖ =

↗ =

-0.5