

Nome: Julia Galger Data: 04/06/2024

1. Analise o código abaixo e responda as questões:

```
1 #include<stdio.h>
2
3 int codigoProva(int valor) {
4     int a = 0;
5     while (valor > 0) {
6         int digito = valor % 10;
7         a += digito;
8         valor /= 10;
9     }
10    return a;
11 }
12
13
14 int main() {
15     int numero;
16
17     printf("Digite um número inteiro positivo: ");
18     scanf("%d", &numero);
19
20     int retornoFunc = codigoProva(numero);
21     printf("Saída do programa: %d\n", retornoFunc);
22
23     return 0;
24 }
```

57.10

50<sup>70</sup>  
0,5

0,5  
a. O código funciona corretamente? Justifique.

Sim, possui a biblioteca, a main com "return 0" no final, a função declarada como int, podendo dessa forma fazer o retorno de um inteiro e ponto e vírgula no final de todas as linhas.

b. Qual é o objetivo principal do programa?

O objetivo do programa é criar um código prova através de um número inteiro. Somar os dígitos de um valor.

30  
2. Um turista chegou a uma casa de câmbio querendo trocar seu dinheiro, deparou-se com um funcionário novo com pouca prática na conversão de moedas. Para ajudá-lo, crie uma função para a fazer a conversão de reais R\$ para uma das outras moedas sem que ele "passe a perna" no cliente, nem em seu chefe. A função deve receber um valor monetário em reais e o código ('e' para estaleca, 'c' para cartoleta) da moeda para conversão. Para isso sabe-se que: CT\$ 1,00 = R\$ 4,00 e E\$ 1,00 = R\$ 2,00. O valor convertido deve ser apresentado.

```
void conversao_valor(int valor_reais, char codigo) {
    float novo_valor;

    if (codigo == 'e') {
        novo_valor = valor_reais / 2;
    }
    else if (codigo == 'c') {
        novo_valor = valor_reais / 4;
    }
    else { printf("Codigo invalido!"); }
    printf("valor convertido: %.f", novo_valor);
}
```

Nome: Julia Jaeger

Data: 04/06/2024

3. Dados os códigos em linguagem de programação C a seguir:

	CÓDIGO 1	CÓDIGO 2
1	#include <stdio.h>	#include <stdio.h>
2		
3	int fnc(int p, int s, int c)	int fnc(int <sup>3</sup> p, int <sup>6</sup> s, int <sup>13</sup> c)
4	{	{
5	int h, a;	int h, a;
6	a = c % 2;	a = c % 2; 1
7	h = (a < 1) ? s / p : (a + 1) % 2;	switch(a)
8	return (h);	{ case 0: { h = s / p; break; }
9	}	case 1: { h = (a + 1) % 2; break; }
10		h = (1+1) % 2
11	int main()	return (h); h = 0
12	{	}
13	int E, F, G, H;	
14	scanf("%d %d", &E, &F);	int main()
15	G = (E / 3) + (2 * F); G = 10	{
16	H = fnc(E, F, G);	int E, F, G, H;
17	E = E + 3;	scanf("%d %d", &E, &F);
18	printf("Valores: %d e %d", H, E);	G = (E / 3) + (2 * F); G = 13
19	return 0;	H = fnc(E, F, G);
20	}	E = E + 3; E = 6
21		printf("Valores: %d e %d", H, E);
22		return 0;
23		}

Análise os itens abaixo e assinale a(s) opção(ões) CORRETA(S), JUSTIFICANDO todos os itens independente de ser correto ou incorreto:

- I. (✓) No código 1, dados os valores 6 e 4 para E e F (na linha 14), respectivamente, o valor de G ao final da execução será 10.
- II. (✗) Os códigos 1 e 2 em linguagem de programação C não apresentam desvios condicionais.
- III. (✗) No código 2, dados os valores 3 e 6 para E e F (na linha 17), respectivamente, o texto apresentado ao usuário no final da execução será: *Valores: 1 e 3*
- IV. (✓) Os códigos 1 e 2 em linguagem de programação C não apresentam estruturas de repetição.
- V. (✗) Os códigos 1 e 2 são algoritmos para problemas diferentes.

- I → O valor de G é o valor de E (6) dividido por 3 =  $\lfloor \frac{6}{3} \rfloor + 2(4) = 2 + 8 = 10$ , então  $2 + 2$ , logo  $G = 10$ .
- II → Falso, o código 1 apresenta o operador Ternário na linha 7 e o código 2 o switch case da linha 7 até a linha 10.
- III → Os valores ao final da execução, será 0 e 6, pois ao executar a função o valor de F fica  $2 \times 2$ , o que equivale a 4 e antes do printf, o E é acrescido de 3, ficando igual a 6.  
↳ Linha 9, por causa do case = 1.
- IV → Em nenhum dos códigos a presença de for, while ou do while.
- V → Ambos os códigos tem a mesma função, a única diferença é o desvio condicional usado.

Nome: Julia Jager

Data: 04/06/2024

4. Analise o código a seguir e responda:

```

1  #include <stdio.h>
2
3  int fnc(int c, int d){
4      if (c % 3 == 0){
5          d=d+1;
6      }
7      return d;
8  }
9
10 int main()
11 {
12     int F=10;
13     int a, b=0, i=0;
14     int G[F]; // vetor[10]
15     while (i < F){
16         scanf("%d", &G[i]); // monta o vetor.
17         i++;
18     }
19     a=G[0]; // a = 1ª posição do vetor. / a=1
20     for(i=0; i < F; i++)
21     {
22         b=fnc(G[i], b);
23         if ( G[i] < a ){
24             a=G[i];
25         }
26     }
27     printf("Saída: %d, %d", b, a);
28     return 0;
29 }

```

1 2 3 4 5 6 7 8 9 10  
 vetor: 0 1 2 3 4 5 6 7 8 9

i=0, c=1, d=0, return 0

i=1, c=2, d=0, return 0

i=0, G[0]=1, b=0

i=1, G[1]=2, b=0

a. Ao final da execução, o que será mostrado na linha 27? Justifique sua resposta:

A linha 27 apresentará os valores das variáveis b e a:

✓ Variável b = Quantos números do vetor são divisíveis por 3.

Variável a = 10 menor número presente no vetor.

b. O que faz a função "fnc"?

Vê se o valor do vetor enviado é divisível por 3, se sim, aumenta 1

✓ na variável d, ou seja, a variável b, se não, apenas retorna a variável com o mesmo valor que chegou na função.

1.5 5. Escreva um programa que leia um valor monetário em reais e uma função que calcule qual o menor número possível de notas de 100, 50, 20, 10, 5, 2 e 1 real(is) que o valor lido pode ser decomposto. Escreva o valor lido e a relação de notas necessárias, bem como a quantidade de cada nota. verro

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

```
int main(){
```

```
    int valor_monetario;
```

```
    scanf("%d", &valor_monetario);
```

```
    numero_notas(valor_monetario);
```

```
    return 0;
```

```
}
```

```
void numero_notas (int valor_reais) {
```

```
int notas_100=0, notas_50=0, notas_20=0, notas_10=0, notas_5=0, notas_2=0, moedas=0;
```

```
int aux = valor_reais;
```

```
int total_notas;
```

```
if (aux > 100) {
```

```
while (aux >= 100) {
```

```
notas_100++;
```

```
aux = aux - 100;
```

```
}
```

```
if (aux >= 50) {
```

```
while (aux >= 50) {
```

```
notas_50++;
```

```
aux = aux - 50;
```

```
}
```

```
if (aux
```

```
while (aux >= 20) {
```

```
notas_20++;
```

```
aux = aux - 20;
```

```
}
```

```
while (aux >= 10) {
```

```
notas_10++;
```

```
aux = aux - 10;
```

```
}
```

```
while (aux >= 5) {
```

```
notas_5++;
```

```
aux = aux - 5;
```

```
}
```

```
while (aux >= 2) {
```

```
notas_2++;
```

```
aux = aux - 2;
```

```
}
```

```
while (aux >= 1) {
```

```
moedas++;
```

```
aux = aux - 1;
```

```
}
```

```
while (aux > 0) {
```

```
if (aux >= 100) {
```

```
notas_100++;
```

```
aux = aux - 100;
```

```
}
```

```
Else if (aux >= 50) {
```

```
notas_50++;
```

```
aux = aux - 50;
```

```
}
```

```
Else if (aux >= 20) {
```

```
notas_20++;
```

```
aux = aux - 20;
```

```
}
```

```
Else if (aux >= 10) {
```

```
notas_10++;
```

```
aux = aux - 10;
```

```
}
```

```
Else if (aux >= 5) {
```

```
notas_5++;
```

```
aux = aux - 5;
```

```
}
```

```
Else if (aux >= 2) {
```

```
notas_2++;
```

```
aux = aux - 2;
```

```
}
```

```
Else if (aux >= 1) {
```

```
moedas++;
```

```
aux = aux - 1;
```

```
}
```

```
}
```

```
Total_notas = notas_100 + notas_50 + notas_20 + notas_10 + notas_5 +  
notas_2 + moedas;
```

```
printf("Valor de entrada: %.d", valor_reais);
```

```
printf("Sei usada %.d notas de 100, %.d notas de 50, %.d notas de 20, %.d notas de 10, %.d notas de 5,  
%.d notas de 2 e %.d moedas de 1", notas_100, notas_50, notas_20, notas_10, notas_5, notas_2, moedas);
```

```
printf("Total de notas: %.d", total_notas);
```

VETOR!

