1. Faça uma função em C para contar os dígitos de um determinado número usando recursão. Faça um programa principal que leia um número, acione a função e exiba o resultado gerado.

2. Faça uma função para encontrar a soma dos dígitos de um número usando recursão. Faça um programa principal que leia um número, acione a função e exiba o resultado gerado.

3. Faça uma função recursiva que calcula a divisão usando subtrações sucessivas:
   int divisao (int numerador, int denominador)
   Faça um programa principal que leia dois números, acione a função e exiba o resultado gerado.

4. Faça uma função recursiva que calcula o resto da divisão usando subtrações sucessivas:
   int resto (int numerador, int denominador)
   Faça um programa principal que leia dois números, acione a função e exiba o resultado gerado.

5. Faça uma função recursiva que calcula o valor de S da série a seguir para n > 0:

   S = 1/1! + 1/2! + 1/3! + ...+1 /N!

   double serie (int n)

   Faça um programa principal que leia um número, acione a função e exiba o resultado gerado.

6. Explique cada uma das expressões a seguir, indicando a diferença entre elas:

   p++; (*p)++; *(p++);

   Qual informação se refere a expressão *(p+10)?

7. Se o endereço de uma variável *valor* foi atribuído a um ponteiro *valorPtr*, quais alternativas são verdadeiras? Justifique sua resposta.

   a) valor = = &valorPtr
   b) valor = = *valorPtr
   c) valorPtr = = &valor
   d) valorPtr = = *valor

8. Identifique o erro no programa a seguir, de modo que seja exibido o valor 10 na tela.

```
#include <stdio.h>
int main()
{
   int x, *p, **q;
   p = &x;
   q = &p;
   x = 10;
   printf("\n%d\n", &q);
   return(0);
}
```

9. Escreva um programa em C que declare variáveis para armazenar um valor inteiro, um valor real e um caracter. Deve existir no programa ponteiros associados a cada um deles. O programa deve solicitar novos dados para as variáveis e elas devem ser modificadas usando os respectivos ponteiros. Exiba os endereços e os conteúdos de todas as variáveis e ponteiros antes e após a alteração.

10. Observe os dois programas a seguir, Código 1 e Código 2. Qual é a diferença entre eles? Qual é o valor impresso para ptr em cada um dos códigos? Por quê?

Código 1	**Código 2**

int main()
{
int *ptr, i;
ptr = (int *) malloc(sizeof(int));
*ptr = 10;
for(i=0;i<5;i++){
*ptr=*ptr+1;
}
printf("\nptr: %d", *ptr);
free(ptr);
return 0;
}
	int main()
	{
	int *ptr, i;
	ptr = (int *) malloc(sizeof(int));
	*ptr = 10;
	for(i=0;i<5;i++){
	ptr=ptr+1;
	}
	printf("\nptr: %p", ptr);
	free(ptr);
	return 0;
	}