

Aula Prática 7 - Turmas E e F

Para os problemas abaixo, pense em um **algoritmo** adequado e **codifique-o** na linguagem C. Não esqueça do cabeçalho no início do programa e da indentação, além de comentários descrevendo passos do programa e as variáveis (quando necessário), tente evitar o uso de variáveis globais.

1. (2 pontos) **Números primos.** Um número é primo se for maior que 1 e somente for divisível por 1 e por ele mesmo. Faça uma função tipada que recebe um número inteiro como parâmetro e retorna 1 se ele é um número primo e zero, caso ele não seja primo. Implemente um programa que lê um inteiro N do teclado e usando a função anterior, imprime o N -ésimo número primo.

Exemplo de execução

```
Entre o inteiro: 5
0 numero primo na posicao 5 eh: 11
```

2. (2 pontos) **Números perfeitos.** Um número é perfeito se for à soma de todos seus divisores próprios (os divisores diferentes dele mesmo). Faça uma função tipada que recebe um número inteiro como parâmetro e retorna 1 se ele é um número perfeito e zero, caso ele não seja perfeito. Implemente um programa que lê um inteiro N do teclado e usando a função anterior, imprime o N -ésimo número perfeito.

Exemplo de execução

```
Entre o inteiro: 1
0 numero perfeito na posicao 1 eh: 6
```

3. (2 pontos) **Números deficientes.** Um número é deficiente se for maior que a soma de todos seus divisores próprios (os divisores diferentes dele mesmo). Faça uma função tipada que recebe um número inteiro como parâmetro e retorna 1 se ele é um número deficiente e zero, caso ele não seja deficiente. Implemente um programa que lê um inteiro N do teclado e usando a função anterior, imprime o N -ésimo número deficiente.

Exemplo de execução

```
Entre o inteiro: 3
0 numero deficiente na posicao 3 eh: 4
```

4. (2 pontos) **Números amigáveis.** Dois números são amigáveis se pelo menos um deles for igual à soma dos divisores próprios do outro. Faça uma função tipada que recebe dois inteiros como parâmetros e retorna 1 se eles forem amigáveis e zero, caso contrário. Implemente um programa que pergunta ao usuário se deseja sair e, enquanto o usuário responda não, leia dois números inteiros e imprima se são amigáveis ou não.

Exemplo de execução

```
Deseja sair? Nao
Entre os numeros: 7 8
Numeros amigaveis!
Deseja sair? Sim
```

5. (1 ponto) Números simpáticos. Um número natural N é simpático se existem inteiros: a , b e c tais $N = a^2 + b^2 - c^2$. Faça uma função tipada que recebe três inteiros como parâmetros e retorna 1 se eles formam um número simpático e zero, caso contrário. Implemente um programa que pergunta ao usuário se deseja sair e, enquanto o usuário responda não, leia três números inteiros e imprima se eles criam um número simpático ou não.

Exemplo de execução

```
Deseja sair? Nao
Entre os numeros: 2 3 4
Nao criam um numero simpatico!
Deseja sair? Sim
```

6. (1 ponto) Números Pitagóricos. Três números são Pitagóricos se cumprem a equação de Pitágoras: o quadrado de um deles deve ser igual à soma dos quadrados dos outros dois. Faça uma função tipada que recebe três inteiros como parâmetros e retorna 1 se eles forem Pitagóricos e zero, caso contrário. Implemente um programa que pergunta ao usuário se deseja sair e, enquanto o usuário responda não, leia três números inteiros e imprima se eles são Pitagóricos ou não.

Exemplo de execução

```
Deseja sair? Nao
Entre os numeros: 3 4 5
Sao Pitagoricos!
Deseja sair? Sim
```

7. (3 pontos) Números Regulares. Um número é regular se a sua decomposição em fatores apresenta apenas potências de 2, 3 e 5. Faça uma função tipada que recebe um número inteiro como parâmetro e retorna 1 se ele é um número regular e zero, caso ele não seja regular. Implemente um programa que pergunta ao usuário se deseja sair e, enquanto o usuário responda não, leia um número inteiro e imprima se ele é regular ou não.

Exemplo de execução

```
Deseja sair? Nao
Entre o numero: 60
Numero regular!
Deseja sair? Sim
```

8. (4 pontos) Número invertido. Dado um número inteiro, seu valor invertido é ele lido de trás para frente. Faça uma função tipada que recebe um número inteiro como parâmetro e retorna o valor invertido. Implemente um programa que pergunta ao usuário se deseja sair e, enquanto o usuário responda não, leia um número inteiro e imprima seu valor invertido.

Exemplo de execução

```
Deseja sair? Nao
Entre o numero: 1234
O numero invertido eh: 4321
Deseja sair? Sim
```

9. (4 pontos) Imprime triângulo. Faça uma função sem retorno que lê um número inteiro que representa o tamanho da base de um triângulo e desenha um triângulo invertido na tela, usando caracteres '*'. Implemente um programa que pergunta ao usuário se deseja sair e, enquanto o usuário responda não, leia um número inteiro e imprima o triângulo associado a ele.

Exemplo de execução

```
Deseja sair? Nao
Entre o tamanho da base: 5
*****
***
*
Deseja sair? Sim
```

10. (2 pontos) Números triangulares. Um número é triangular se ele é igual ao produto de três inteiros consecutivos. Faça uma função tipada que recebe um número inteiro como parâmetro e retorna 1 se ele é um número triangular e zero, caso ele não seja triangular. Implemente um programa que pergunta ao usuário se deseja sair e, enquanto o usuário responda não, leia um número inteiro e imprima se ele é triangular ou não.

Exemplo de execução

```
Deseja sair? Nao
Entre o numero: 24
Eh triangular!
Deseja sair? Sim
```

11. (4 pontos) Persistência multiplicativa. A persistência multiplicativa de um número inteiro é o número de vezes que os dígitos deste número devem ser multiplicados, para que o resultado chegue a um único dígito. Faça uma função tipada que recebe um número inteiro como parâmetro e retorna a sua persistência multiplicativa. Implemente um programa que pergunta ao usuário se deseja sair e, enquanto o usuário responda não, leia um número inteiro e imprima sua persistência multiplicativa.

Exemplo de execução

```
Deseja sair? Nao
Entre o numero: 6423
Persistencia multiplicativa: 3
Deseja sair? Sim
```

12. (3 pontos) Ponto no plano. Faça uma função sem retorno que recebe dois números inteiros entre 0 e 19 que representam as coordenadas x e y de um ponto em um plano bidimensional 20×20 , cuja origem fica no canto inferior esquerdo. A função deve desenhar este plano na tela, usando o caractere " ", e indicando a posição do ponto com o caractere 'P'. Implemente um programa que pergunta ao usuário se deseja sair e, enquanto o usuário responda não, leia dois números inteiros que representem o ponto e imprima o plano correspondente.

Exemplo de execução

```
Deseja sair? Nao
Entre as coordenadas do ponto: 5 8
#####
#####
#####
#####
#####
#####
#####
```


#####P#####

#####

Deseja sair? Sim