

Primeira lista de exercícios v1.00
Estatística - 2º Semestre de 2015
Profs: Daniel Rodrigo Ferraz Bonetti
PAE: Daniel Tozadore

1. Faça um programa que imprima um triângulo isóscele à direita de altura n, digitada pelo usuário. Ex:

Entrada: n = 3

Saída:

```
*  
**  
***
```

2. Faça um programa que imprima um triângulo de lado, com altura $2n-1$ e com largura n.

Ex:

Entrada: n = 4

Saída:

```
*  
**  
***  
****  
***  
**  
*
```

3. Faça um programa que imprima um triângulo de altura n e com $2n-1$ de largura. Ex:

Entrada: n = 6

Saída:

```
  *  
 ***  
*****  
*****  
*****  
*****  
*****
```

4. Faça uma função que verifique se um inteiro é positivo; a declaração da função deve ser: `int isPositive(int i)`, e deverá retornar 0 se for negativo e 1 caso contrário.

5. Implemente uma função que calcula x^y (não usar as funções prontas, você deve desenvolver a lógica para o cálculo da potência). Você deve utilizar a função da seguinte maneira:

```
float a = raiseToPower(2, 3); // retorna 8
```

```
float b = raiseToPower(9, 2); // retorna 81
```

```
float raiseToPower(float f, int power); // declaração da sua função
```

6. Implemente uma função que verifica se um número é primo. Essa função deverá retornar 1 se o número for primo ou 0, caso contrário. Mostrar na tela se o número é ou não primo. Ex:

Entrada: 11

Saída: O número 11 é um número primo

7. Faça uma função que determina quantos números são primos abaixo de um valor n digitado pelo usuário. Ex:

Entrada: 12

Saída: Existem 5 números primos menores que 12.

8. Crie uma função que receba três valores, 'a', 'b' e 'c', que são os coeficientes de uma equação do segundo grau e retorne o valor do delta, que é dado por ' $b^2 - 4ac$ '

9. Utilize as funções dos Exercícios 4 e 8 para criar um programa que calcula as raízes de uma equação do 2º grau:

$$ax^2 + bx + c = 0$$

Para ela existir, o coeficiente 'a' deve ser diferente de zero.

Caso o delta seja maior ou igual a zero, as raízes serão reais. Caso o delta seja negativo (função do Exercício 4), as reais serão complexas e da forma: $x + iy$

10. Crie uma função que receba um vetor inteiro e o tamanho do vetor por parâmetro e calcule e retorne a média desse vetor. Faça outra função que receberá um vetor de floats.

11. Utilizando a função do Exercício 10 (para variáveis reais), faça uma função que calcule o desvio padrão de um vetor.

12. Implemente uma função que receba dois valores e retorne o MDC (Máximo Divisor Comum) entre eles. Faça outra função que retorne o MMC (Mínimo Múltiplo Comum) entre dois números.

13. Crie um programa em C que receba números inteiros digitados pelo usuário. Quando o usuário digitar um número negativo, o programa deve retornar a média (Exercício 10) e o desvio padrão (Exercício 11) dos números positivos digitados anteriormente.

14. Crie um programa de conversão entre as temperaturas Celsius e Fahrenheit. Primeiro o usuário deve escolher se vai entrar com a temperatura em Celsius ou Fahrenheit, depois a conversão escolhida é realizada através de um comando SWITCH.

Se C é a temperatura em Celsius e F em Fahrenheit, as fórmulas de conversão são:

$$C = 5 \cdot (F - 32) / 9$$

$$F = (9 \cdot C / 5) + 32$$

15. Crie uma função em linguagem C chamado Dado() que retorna, através de sorteio, um número de 1 até 6.

16. Use a função do Exercício 15 e lance o dado 1 milhão de vezes. Conte quantas vezes cada número saiu. A probabilidade deu certo? Ou seja, a porcentagem dos números foi parecida?

17. Faça um programa que leia três notas e dê o conceito de acordo com as regras:

-se o conceito médio $\geq 90\%$ \Rightarrow conceito=A

-se o conceito médio $\geq 70\%$ e $< 90\%$ \Rightarrow conceito=B

-se o conceito médio $\geq 50\%$ e $< 70\%$ \Rightarrow conceito=C

-se o conceito médio $< 50\%$ \Rightarrow conceito=F

17. Faça um programa que imprima a seguinte saída a partir de um valor n digitado pelo usuário: Ex:

Entrada: n = 7

Saída:

```
1*****
12*****
123****
1234***
12345**
123456*
1234567
```

18. Faça um programa que solicite ao usuário digitar um valor inteiro positivo. O programa deverá termina quando um número negativo for digitado.

19. Implemente uma calculadora com o seguinte menu:

MENU

1. Adição
2. Subtração
3. Multiplicação
4. Divisão
5. Módulo

Escolha uma opção: 1

Entre com dois números: 12 15

Resultado: 27

Continuar? y

O programa também pergunta ao usuário se quer continuar a execução. Se o usuário digitar 'y' então o programa segue com a execução ou termina caso outro caractere seja digitado.

20. Faça um programa que leia o nome de n países e mostre o país que tem o nome mais comprido digitado.

Dica: utilizar a função strlen da biblioteca string.h

21. Faça um programa que sorteie um valor entre 1 e 100. Depois, peça que o usuário tente adivinhar esse número sugerindo se o número digitado pelo usuário é maior ou menor que número sorteado. O programa terminará depois que o usuário acertar o número. Deverá exibir na tela também o número de tentativas necessárias que o usuário precisou para acertar o número. A estrutura básica do programa pode ser:

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
    // suas variáveis aqui
    int num;
    srand(time(NULL)); // semente aleatória
    num = (rand()%100)+1; // obtém um número aleatório
    // escreva a rotina aqui
    return 0;
}
```

22. Faça uma função chamada Swap que receba dois valores por referência e troque o valor do primeiro com o segundo parâmetro.

23. Faça uma sub-rotina que receba como parâmetros três variáveis inteiras e retorne estas variáveis com seus valores ordenados, ou seja, o menor valor na primeira variável, o segundo menor valor na variável do meio, e o maior valor na última variável. A rotina deve retornar o valor 1 se os três valores forem iguais e 0 se existirem valores diferentes.

Aviso: (opcional) Você pode criar apenas um projeto em sua IDE preferida (Visual Studio, Codeblocks, Dev etc) e, dentro da função main(), criar uma função para cada exercício. Ex:

```
void exer01() {
    // conteúdo do Exercício 01 aqui
}
```

```
void exer02() {
    // conteúdo do Exercício 02 aqui
}
```

```
void exer03() {
    // conteúdo do Exercício 03 aqui
}

.
.
.

void exerN() {
    // conteúdo do Exercício N aqui
}

int main() {

    // exer01(); // Você pode comentar os exercícios que já rodou
    // exer02();
    // exer03();
    ...();
    exerN();
}

Boa sorte ;)
```