Universidade Federal do Ceará
Campus Russas

Laboratório de Programação
Lista de exercícios 2

1.

(a) Faça uma função recursiva que recebe um inteiro $n \geq 0$ e retorna o fatorial $n!$. Lembre-se que $0! = 1$.

(b) Utilizando a função do item anterior, faça uma função que recebe dois inteiros $n \geq 0$ e $m \geq 0$ (onde $n \geq m$) e retorna a combinação de n por m que é $\binom{n}{m} = \frac{n!}{m!(n-m)!}$.

(c) Faça uma função que recebe um inteiro k e imprime os valores até a $(k+1)$-ésima linha do Triângulo de Pascal.

$$
\begin{array}{ccccccccc}
\binom{0}{0} & & & & & & & & \\
\binom{1}{0} & \binom{1}{1} & & & & & & & \\
\binom{2}{0} & \binom{2}{1} & \binom{2}{2} & & & & & & \\
\binom{3}{0} & \binom{3}{1} & \binom{3}{2} & \binom{3}{3} & & & & & \\
\binom{4}{0} & \binom{4}{1} & \binom{4}{2} & \binom{4}{3} & \binom{4}{4} & & & & \\
\binom{5}{0} & \binom{5}{1} & \binom{5}{2} & \binom{5}{3} & \binom{5}{4} & \binom{5}{5} & & & \\
\binom{6}{0} & \binom{6}{1} & \binom{6}{2} & \binom{6}{3} & \binom{6}{4} & \binom{6}{5} & \binom{6}{6} & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \\
\binom{k}{0} & \binom{k}{1} & \binom{k}{2} & \binom{k}{3} & \binom{k}{4} & \binom{k}{5} & \binom{k}{6} & \cdots & \binom{k}{k}
\end{array}
$$

2.

(a) Faça uma função recursiva que recebe um inteiro $n \geq 0$ e retorna o Fibonacci de n.

(b) Faça uma função iterativa que recebe um inteiro $n \geq 0$ e retorna o Fibonacci de n.

(c) Faça um programa que recebe um inteiro $n \geq 0$ e retorna o tempo que cada uma das funções dos itens anteriores. Qual função leva mais tempo quando n é grande? Para responder esse item utilize a função *clock()* da biblioteca *time.h* (veja o exemplo de uso a seguir).

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

int main(){
        clock_t tempo_inicial,tempo_final; //declaração de variáveis clock_t

        printf("Inicio de marcação de tempo do 'for'\n");
        tempo_inicial = clock();

        for(int i=0;i<2147483647;i++); // esse 'for' não faz nada, apena gasta
            tempo

        tempo_final = clock();

        printf("Tempo gasto: %f ms\n\n",1000*(double)(tempo_final-tempo_inicial
            )/(double)(CLOCKS_PER_SEC));

        return 0;
}
```

3.

(a) Faça uma função que recebe um inteiro n e retorna o número de dígito que n possui. Por exemplo, o número 1957862 possui 7 dígitos.

(b) Faça uma função que recebe um inteiro n e inteiro $d \in [0, 9]$ que retorna o número de dígitos d que o número n possui. Por exemplo $n = 45647$ e $d = 4$ o programa deve retorna 2, pois 45647 possui dois dígitos 4.

(c) Um número x é uma **permutação** de um número y se os dígitos de y são uma permutação dos dígitos de x (ex. 45647 é permutação de 65447). Utilizando a função da questão anterior, faça uma função que recebe dois inteiros $x > 0$ e $y > 0$, onde nenhum de seus dígitos seja 0, e retorna 1 se x é permutação de y, 0 em caso contrário.

(d) Um número x é dito **sufixo** de um número y se os dígitos de x são últimos dígitos de y (ex. 654 é sufixo de 789654). Faça uma função recebe dois inteiro $x > 0$ e $y > 0$ que retorna 1 se x é sufixo de y, 0 caso contrário.

(e) Utilizando a função do item anterior, faça uma função que recebe dois inteiros $x > 0$ e $y > 0$ e retorna 1 se o menor dos números é um segmento do outro, 0 em caso contrário. Por exemplo, 7561 é um segmento do número 98**7561**45.

4. Um número é um **palíndromo** quando é o mesmo de trás para frente (ex. 123321 é um palíndromo enquanto 1231 não é). Faça uma função que recebe um número inteiro x e retorna 1 se x é um palíndromo, 0 em caso contrário.

5. Faça uma função que recebe um inteiro n, na base binária, e retorna o valor desse inteiro na base decimal. Por exemplo, para $n = 101$ deve retornar 5.