lista 5 ap2

1)

```
#include <stdio.h>

int NATURAIS(int n)
{
   if (n == 0) // caso base
   {
      return 1;
   }
   else
   {
      return n + NATURAIS(n - 1); // nesse caso n é igual a 9 então, 9 + NATURIAIS(9-1)= 9+8=17, depois 8 + Naturais(8-1)= 8+7= 15, e assim por diante até que chegue no caso base.
   }
}

int main()
{
   int n = 9;
   printf("%i", NATURAIS(n)); // aqui a função NATURAIS copia o valor de n para si.

   return 0;
}
```

2)

```
#include <stdio.h>

int MULT(int a, int b)
{
    if (a == 1 && b == 1)
    {
        return 1; // caso base (critério de parada)
    }
    else if(a>b)
    {
        return a + MULT(a -1); // 10+10+10+10+10
                               //10+10
    }
}

int main()
{
    int a, b;
```

```
    printf("digite dois valores: ");
    scanf("%i %i", &a, &b);

    MULT(a, b);

    return 0;
}
#include <stdio.h>

int MULT(int a, int b)
{
    if (a == 1 && b == 1)
    {
        return 1; // caso base (critério de parada)
    }
    else if(a>b)
    {
        return a + MULT(a -1); // 10+10+10+10+10
                               //10+10
    }
}

int main()
{
    int a, b;

    printf("digite dois valores: ");
    scanf("%i %i", &a, &b);

    MULT(a, b);

    return 0;
}
```

3)

```
#include <stdio.h>

int POT(int x, int n)
{
    if (n == 0)//caso base
    {
        return 1; // retorna 1 pois x^0 = 1
    }
```

```
    else
    {
        return x * POT(x, n - 1);
    }
}
int main()
{
    int x, n;

    printf("Digite um valor para x e n: ");
    scanf("%i %i", &x, &n);

    POT(x, n);

    printf("%i", POT(x, n));

    return 0;
}
```

4)

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

int fatorial(int n) // para calcular o fatorial
{
    if (n == 0 || n == 1)
    {
        return 1;
    }
    else
    {
        return n * fatorial(n - 1);
    }
}

int FATQUADRU(int n)
{
    return fatorial(2 * n) / fatorial(n);
}

int main()
{
    setlocale(LC_ALL, "pt_BR.UTF-8");
```

```
    int n;
    printf("digite o valor de n: ");
    scanf("%i", &n);

    int m = FATQUADRU(n);

    printf("O fatorial quádruplo de %i é: %i\n", n, m);

    return 0;
}
```

5)

```
#include <stdio.h>

int FIBONACCI(int n)
{
    if (n == 0) // caso base
    {
        return 0;
    }
    else if (n == 1) // caso base
    {
        return 1;
    }
    else // caso recursivo
    {
        return FIBONACCI(n - 1) + FIBONACCI(n - 2);
    }
}

int main()
{
    int n;

    printf("digite um valor para n: ");
    scanf("%i", &n);

    int m = FIBONACCI(n);

    printf("Sequência de Fibonacci: %i ", m);
```

```
    return 0;
}
```

6) #include <stdio.h>

```
#include <stdlib.h>

int SOMATORIO(int n)
{
    if (n == 1) // caso base

        return 0;
    else
        return 2 * (n - 1) + SOMATORIO(n - 1); // se por exemplo n for iguas a 6 ->  2 *(6-1) =15 , 15 soma ao caso recursivo -> somatorio(n-1) -> n-1 = 6-1= 5, 5-1=4, 4-1=3, 3-1=2, 2-1=1,1-1= caso base. os resultados 5,4,3,2,1 são somados ao numero 15, resultando em 30.
}

int main()
{
    int n;

    printf("digite o valor de n: ");
    scanf("%i", &n);

    int m = SOMATORIO(n);

    printf("%i", m);
    return 0;
}
```

7)#include <stdio.h>

```

int TRIBONACCI(int n)
{
    if (n == 0)
    {
        return 0;
    }
    else if (n == 1)
```

```
    {
        return 0;
    }
    else if (n == 2)
    {
        return 1;
    }
    else if (n > 2)
    {
        return TRIBONACCI(n - 1) + TRIBONACCI(n - 2) + TRIBONACCI(n -
3);
    }
}
int main()
{
    int n;

    printf("digite um valor para n: ");
    scanf("%i", &n);

    int result = TRIBONACCI(n);

    printf("resultado: %i", result);
    return 0;
}
```

8)

```
#include <stdio.h>

int NUMPELL(int n)
{
    if (n == 0)
    {
        return 0;
    }
    else if (n == 1)
    {
        return 1;
    }
    else if (n > 2)
    {
        return 2 * NUMPELL(n - 1) + NUMPELL(n - 2);
    }
```

```c
}
int main()
{
    int n, m;

    printf("digite um valor para n: ");
    scanf("%i", &n);

    m = NUMPELL(n);

    printf("%i", m);

    return 0;
}
```