

Tarefa Recursão

Exercício 1

```
1  #include <stdio.h>
2
3  int fat(int n) {
4      if (n == 0) {
5          return 1;
6      }
7
8      return n * fat(n - 1);
9  }
10
11 int main(void) {
12     int n;
13     printf("Número? ");
14     scanf("%d", &n);
15     printf("Fat = %d\n", fat(n));
16     return 0;
17 }
```

Saída:

```
.../Recursão/exercicios > ./output/*1
Numero? 5
Fat = 120
```

Exercício 2

```
1  #include <stdio.h>
2
3  float pot(float x, int n) {
4      if (n == 0) {
5          return 1;
6      }
7
8      return x * pot(x, n - 1);
9  }
10
11 int main(void) {
12     float x;
13     int n;
14
15     printf("Base e expoente? ");
```

```

16     scanf("%f %d", &x, &n);
17     printf("%.1f^%d = %.1f", x, n, pot(x, n));
18     return 0;
19 }

```

Saída:

```

.../Recursão/exercicios > ./output/exercicio2
Base e expoente? 2 4
2.0^4 = 16.0
.../Recursão/exercicios > ./output/exercicio2
Base e expoente? 2 5
2.0^5 = 32.0

```

Exercício 3

```

1     #include <stdio.h>
2
3     int termial(int n) {
4         if (n == 0)
5             return 0;
6
7         return termial(n - 1) + n;
8     }
9
10    int main() {
11        int n;
12
13        printf("Número? ");
14        scanf("%d", &n);
15        printf("Termial = %d", termial(n));
16
17        return 0;
18    }

```

Saída:

```

.../Recursão/exercicios > ./output/exercicio3
Número? 5
Termial = 15
.../Recursão/exercicios > ./output/exercicio3
Número? 10
Termial = 55

```

Exercício 4

```

1     #include <stdio.h>
2     float h(int n) {
3         if (n == 1)
4             return 1;

```

```

5
6     return h(n - 1) + 1.0 / n;
7 }
8
9 int main() {
10     int n;
11     printf("Número? ");
12     scanf("%d", &n);
13     printf("Harmonica = %.2f\n", h(n));
14     return 0;
15 }

```

Saída:

```

.../Recursão/exercicios > ./output/*4
Número? 5
Harmonica = 2.28

.../Recursão/exercicios > ./output/*4
Número? 4
Harmonica = 2.08

```

Exercício 5

```

1     #include <stdio.h>
2
3     void cp(int n) {
4         if (n == 0)
5             return;
6
7         cp(n - 1);
8         printf("%d\n", n);
9     }
10
11    int main() {
12        int n;
13        printf("Contagem progressiva. diga o número para contagem\n\número = ");
14        scanf("%d", &n);
15        cp(n);
16        return 0;
17    }

```

Saída:

```

.../Recursão/exercicios } ./output/exercicio5
Contagem progressiva. diga o número para contagem
número = 10
1
2
3
4
5
6
7
8
9
10

```

Exercício 6

```

1  #include <stdio.h>
2
3  void cr(int n) {
4      if (n == 0)
5          return;
6
7      printf("%d\n", n);
8      cr(n - 1);
9  }
10
11 int main(void) {
12     int n;
13     printf("Contagem regressiva. De qual número quer contar?\nNúmero = ");
14     scanf("%d", &n);
15     cr(n);
16     return 0;
17 }

```

```

.../Recursão/exercicios } ./output/exercicio6
Contagem regressiva. De qual número quer contar?
Número = 10
10
9
8
7
6
5
4
3
2
1

```

Saída:

Exercício 7

```
1  #include <stdio.h>
2
3  void binario(int n) {
4      if (n < 2)
5          printf("%d", n);
6      else {
7          binario(n / 2);
8          printf("%d", n % 2);
9      }
10 }
11
12 int main() {
13     int n;
14     printf("Qual o número? ");
15     scanf("%d", &n);
16     binario(n);
17     return 0;
18 }
```

Saída:

```
.../Recursão/exercicios > ./output/exercicio7
Qual o número? 13
1101
```

Exercício 8

```
1  #include <stdio.h>
2
3  void hanoi(int n, char origem, char auxiliar, char destino) {
4      if (n == 1) {
5          printf("Mova o disco 1 de %c para %c\n", origem, destino);
6          return;
7      }
8      hanoi(n - 1, origem, destino, auxiliar);
9      printf("Mova o disco %d de %c para %c\n", n, origem, destino);
10     hanoi(n - 1, auxiliar, origem, destino);
11 }
12
13 int main() {
14     int n;
15     printf("Quantos discos? ");
16     scanf("%d", &n);
17     hanoi(n, 'A', 'B', 'C');
18     return 0;
19 }
```

Saída:

```

.../Recursão/exercicios } ./output/exercicio8
Quantos discos? 3
Mova o disco 1 de A para C
Mova o disco 2 de A para B
Mova o disco 1 de C para B
Mova o disco 3 de A para C
Mova o disco 1 de B para A
Mova o disco 2 de B para C
Mova o disco 1 de A para C

```

Exercício 9

```

1  #include <stdio.h>
2
3  int par(int n) {
4      if (n == 0) {
5          return 0;
6      } else if (n == 1) {
7          return 1;
8      } else {
9          return par(n - 2);
10         if (n == 1) {
11             return 1;
12         }
13         return 0;
14     }
15 }
16
17 int main(void) {
18     int n;
19     printf("Digite um número: ");
20     scanf("%d", &n);
21
22     if (par(n) == 0) {
23         printf("Digito é par");
24     } else {
25         printf("Digito é impar");
26     }
27     return 0;
28 }

```

Saída:

```

.../Recursão/exercicios } ./output/exercicio9
Digite um número: 8
Digito é par
.../Recursão/exercicios } ./output/exercicio9
Digite um número: 7
Digito é impar

```

Exercício 10

```

1  #include <stdio.h>
2
3  int prod(int m, int n) {
4      if (n == 0) {
5          return 0;
6      }
7
8      return m + prod(m, n - 1);
9  }
10
11 int main(void) {
12     int num1;
13     int num2;
14     printf("Numeros a ser multiplicados?\n");
15     while (1) {
16         scanf("%d %d", &num1, &num2);
17         if (num1 < 0 || num2 < 0) {
18             printf("Só pode ser numero positivo!\n");
19         } else {
20             printf("%d * %d = %d", num1, num2, prod(num1, num2));
21             break;
22         }
23     }
24 }

```

Saída:

```

.../Recursão/exercicios > ./output/exercicio10
Numeros a ser multiplicados?
9 -7
Só pode ser numero positivo!
8 9
8 * 9 = 72

```

Exercício 11

```

1  #include <stdio.h>
2
3  int quoc(int m, int n) {
4      if (m < n) {
5          return 0;
6      }
7
8      return 1 + quoc(m - n, n);
9  }
10

```

```

11  int main(void) {
12      int n1, n2;
13
14      while (1) {
15          printf("Qual o Dividendo? ");
16          scanf("%d", &n1);
17          printf("Qual o Divisor? ");
18          scanf("%d", &n2);
19
20          if (n1 < 1 || n2 < 1) {
21              printf("Divisão não autorizada");
22          } else {
23              printf("%d / %d = %d", n1, n2, quoc(n1, n2));
24              break;
25          }
26      }
27
28      return 0;
29  }

```

Saída:

```

../Recursão/exercicios > ./output/exercicio11
Qual o Dividendo? 15
Qual o Divisor? 4
15 / 4 = 3

```

Exercício 12

```

1  #include <stdio.h>
2
3  int quoc(int m, int n) {
4      if (m < n) {
5          return 0;
6      }
7
8      return 1 + quoc(m - n, n);
9  }
10
11 int resto(int m, int n) {
12     if (m < n) {
13         return m;
14     }
15
16     return resto(m - n, n);
17 }
18
19 int main(void) {
20     int n1, n2;

```



```

21
22     while (1) {
23         printf("Qual o Dividendo? ");
24         scanf("%d", &n1);
25         printf("Qual o Divisor? ");
26         scanf("%d", &n2);
27
28         if (n1 < 1 || n2 < 1) {
29             printf("Divisão não autorizada");
30         } else {
31             printf("%d / %d = %d\\ne o resto = %d", n1, n2, quoc(n1, n2),
32                 resto(n1, n2));
33             break;
34         }
35     }
36
37     return 0;
38 }

```

Saída:

```

../Recursão/exercicios > ./output/exercicio12
Qual o Dividendo? 15
Qual o Divisor? 4
15 / 4 = 3
e o resto = 3

```

Exercício 13

```

1     #include <stdio.h>
2
3     int quadrado(int num) {
4         if (num < 1) {
5             return 0;
6         }
7
8         int n = quadrado(num - 1);
9         int count = 2 * num - 1;
10
11        return n + count;
12    }
13
14    int main() {
15        int num;
16        printf("Número? ");
17        scanf("%d", &num);
18        printf("%d^2 = %d", num, quadrado(num));
19        return 0;
20    }

```

Saída:

```

.../Recursão/exercicios > ./output/*13
Número? 5
5^2 = 25
.../Recursão/exercicios > ./output/*13
Número? 10
10^2 = 100
.../Recursão/exercicios >

```

Exercício 14

```

1  #include <stdio.h>
2
3  int sd(int n) {
4      if (n == 0)
5          return 0; // Caso base
6
7      return (n % 10) + sd(n / 10); // Passo recursivo
8  }
9
10 int main(void) {
11     int n;
12     printf("Digite um número: ");
13     scanf("%d", &n);
14
15     printf("A soma dos dígitos de %d é %d\n", n, sd(n));
16
17     return 0;
18 }

```

Saída:

```

.../Recursão/exercicios x ./output/exercicio14
Digite um número: 7859
A soma dos dígitos de 7859 é 29

```

Exercício 15

```

1  #include <stdio.h>
2
3  int qd(int n) {
4      if (n < 2)
5          return 1;
6
7      return 1 + qd(n / 2);
8  }
9

```

```

10 void binario(int n) {
11     if (n < 2)
12         printf("%d", n);
13     else {
14         binario(n / 2);
15         printf("%d", n % 2);
16     }
17 }
18
19 int main() {
20     int n;
21     printf("Qual o número? ");
22     scanf("%d", &n);
23     printf("A quantidade de digitos binários para o %d é: %d\n", n, qd(n));
24     printf("Pois %d em binário é: ", n);
25     binario(n);
26     return 0;
27 }

```

Saída:

```

.../Recursão/exercicios > ./output/exercicio15
Qual o número? 13
A quantidade de digitos binários para o 13 é: 4
Pois 13 em binário é: 1101

```

Exercício 16

```

1  #include <stdio.h>
2
3  int h(int n) {
4      if (n == 1)
5          return 1;           // Caso base: 1 disco precisa de 1 movimento
6      return 2 * h(n - 1) + 1; // Passo recursivo
7  }
8
9  void hanoi(int n, char origem, char auxiliar, char destino) {
10     if (n == 1) {
11         printf("Mova o disco 1 de %c para %c\n", origem, destino);
12         return;
13     }
14     hanoi(n - 1, origem, destino, auxiliar);
15     printf("Mova o disco %d de %c para %c\n", n, origem, destino);
16     hanoi(n - 1, auxiliar, origem, destino);
17 }
18
19 int main() {
20     int n;
21     printf("Quantos discos? ");

```

```

22     scanf("%d", &n);
23
24     printf("\nNúmero mínimo de movimentos: %d\n\n", h(n));
25     hanoi(n, 'A', 'B', 'C');
26
27     return 0;
28 }

```

Saída:

```

.../Recursão/exercicios > ./output/exercicio16
Quantos discos? 3

Número mínimo de movimentos: 7

Mova o disco 1 de A para C
Mova o disco 2 de A para B
Mova o disco 1 de C para B
Mova o disco 3 de A para C
Mova o disco 1 de B para A
Mova o disco 2 de B para C
Mova o disco 1 de A para C

```

Exercício 17

```

1  #include <stdio.h>
2  #include <string.h>
3
4  char *inv(char s[], int p, int u) {
5      if (p >= u) {
6          return s; // caso base: já inverteu tudo
7      }
8
9      // troca os caracteres nas posições p e u
10     char temp = s[p];
11     s[p] = s[u];
12     s[u] = temp;
13
14     // chama recursivamente com os índices internos
15     return inv(s, p + 1, u - 1);
16 }
17
18 int main() {
19     char s[100];
20     printf("digite uma string: ");
21     scanf("%s", s);
22
23     int tamanho = strlen(s);
24     printf("string invertida: %s\n", inv(s, 0, tamanho - 1));

```

```

25
26     return 0;
27 }

```

Saída:

```

.../Recursão/exercicios ) ./output/exercicio17
Digite uma string: teste
String invertida: etset

```

Exercício 18

```

1  #include <ctype.h>
2  #include <stdio.h>
3  #include <string.h>
4
5  int pal(char s[], int p, int u) {
6      while (p < u && s[p] == ' ')
7          p++;
8      while (u > p && s[u] == ' ')
9          u--;
10
11     if (p >= u)
12         return 1;
13
14     if (tolower(s[p]) != tolower(s[u]))
15         return 0;
16
17     return pal(s, p + 1, u - 1);
18 }
19
20 int main() {
21     char s[100];
22     printf("Digite uma frase: ");
23     fgets(s, sizeof(s), stdin);
24
25     int tamanho = strlen(s);
26     if (s[tamanho - 1] == '\n')
27         s[tamanho - 1] = '\0';
28
29     if (pal(s, 0, strlen(s) - 1))
30         printf("É palíndromo!\n");
31     else
32         printf("Não é palíndromo.\n");
33
34     return 0;
35 }

```

```
.../Recursão/exercicios > ./output/exercicio18  
Digite uma frase: subi no onibus  
É palíndromo!
```

```
.../Recursão/exercicios > ./output/exercicio18  
Digite uma frase: bora testar  
Não é palíndromo.
```