

Universidade de Itaúna - Ciência da Computação
Laboratório de Algoritmos e Estruturas de Dados I
Professor: Thiago Silva Vilela
Solução de Exercícios sobre Funções

Exercício 1

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define PI 3.14
5
6 float calculaPerimetro(float r) {
7     return 2*PI*r;
8 }
9
10 float calculaArea(float r) {
11     return PI*r*r;
12 }
13
14 int main(int argc, char *argv[]) {
15     float raio;
16     printf("Forneca o raio da circunferencia: ");
17     scanf("%f", &raio);
18     printf("Area: %f\n", calculaArea(raio));
19     printf("Perimetro: %f\n", calculaPerimetro(raio));
20     return 0;
21 }
```

Exercício 2

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int maiorValor(int a, int b) {
5     if (a > b) {
6         return a;
7     } else {
8         return b;
9     }
10 }
11
12 int maiorValor3(int a, int b, int c) {
13     return maiorValor(maiorValor(a, b), c);
14 }
15
16 int main(int argc, char* argv[]) {
17     int x, y, z;
18     printf("Forneca tres valores inteiros: ");
19     scanf("%d %d %d", &x, &y, &z);
20
21     int maior = maiorValor3(x, y, z);
22     printf("O maior valor e: %d\n", maior);
23     return 0;
24 }
```

Exercício 3

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int dado() {
6     return rand()%6+1;
7 }
8
9 int main(int argc, char *argv[]) {
10     srand(time(NULL));
11     // Vamos usar um vetor de 7 posicoes e ignorar o indice 0!
12     // O indice i vai armazenar quantas vezes tivemos o resultado i
13     // no lancamento de um dado.
14     int resultados[7];
15     int i;
16     for (i = 0; i <= 6; i++) {
17         resultados[i] = 0;
18     }
19     for (i = 0; i < 1000000; i++) {
20         resultados[dado()]++;
21     }
22     for (i = 0; i <= 6; i++) {
23         printf("%d: %d\n", i, resultados[i]);
24         resultados[i] = 0;
25     }
26     return 0;
27 }
```

Exercício 4

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int verificaPrimo(int x) {
5     int i, num = 0;
6     if (x == 1) return 0; //O numero 1 nao eh primo!
7     for (i = 1; i <= x; i++) {
8         if (x%i == 0) {
9             num++;
10        }
11        if (num > 2) {
12            return 0;
13        }
14    }
15    return 1;
16 }
17
18 int main(int argc, char *argv[]) {
19     int x, i, soma, proximo_primo;
20     printf("Entre com um numero positivo: ");
21     scanf("%d", &x);
22     soma = 0;
23     i = 0;
24     proximo_primo = 1;
25     while (i < x) {
26         if (verificaPrimo(proximo_primo)) {
27             soma += proximo_primo;
28             i++;
29         }
30         proximo_primo++;
31     }
```

```

32     printf("A soma dos %d primeiros numeros primos eh: %d.\n", x, soma);
33     return 0;
34 }

```

Exercício 5

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int contadigitos(int n, int d) {
5      int conta, resto;
6      conta = 0;
7      while (n != 0) {
8          resto = n % 10;
9          n = n/10;
10         if (resto == d) {
11             conta++;
12         }
13     }
14     return conta;
15 }
16
17 int main(int argc, char *argv[]) {
18     int a, b, i;
19     int perm; /* indicador de passagem: se perm == 1 e' permutacao */
20
21     printf("Digite dois inteiros: ");
22     scanf("%d %d",&a,&b);
23     perm = 1;
24
25     for (i = 1; i <= 9; i++) {
26         if (contadigitos(a,i) != contadigitos(b,i)) {
27             perm = 0;
28         }
29     }
30
31     if (perm == 1) {
32         printf("%d e' permutacao de %d\n",a,b);
33     } else {
34         printf("%d nao e' permutacao de %d\n",a,b);
35     }
36     return 0;
37 }

```