
```
1  // Oportunidade para melhorar o design
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7      printf("atchim\n");
8      printf("atchim\n");
9      printf("atchim\n");
10 }
```

```
1  // Design melhor
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7      for (int i = 0; i < 3; i++)
8      {
9          printf("atchim\n");
10     }
11 }
```

```
1  // Abstração
2
3  #include <stdio.h>
4
5  void atchim(void);
6
7  int main(void)
8  {
9      for (int i = 0; i < 3; i++)
10     {
11         atchim();
12     }
13 }
14
15 // Espirra uma vez
16 void atchim(void)
17 {
18     printf("atchim\n");
19 }
```

```
1  // Abstração usando parâmetros
2
3  #include <stdio.h>
4
5  void atchim(int n);
6
7  int main(void)
8  {
9      atchim(3);
10 }
11
12 // Espirra um número n de vezes
13 void atchim(int n)
14 {
15     for (int i = 0; i < n; i++)
16     {
17         printf("atchim\n");
18     }
19 }
```

```
1 // Operadores lógicos
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Pergunta ao usuário se ele concorda
9     char c = get_char("Você concorda?\n");
10
11     // Checa se usuário concordou
12     if (c == 'S' || c == 's')
13     {
14         printf("Concordou.\n");
15     }
16     else if (c == 'N' || c == 'n')
17     {
18         printf("Não concordou.\n");
19     }
20 }
```

```
1 // Condicionais e operadores relacionais
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Pedir ao usuário para informar x
9     int x = get_int("x: ");
10
11     // Pedir ao usuário para informar y
12     int y = get_int("y: ");
13
14     // Comparar x e y
15     if (x < y)
16     {
17         printf("x é menor do que y\n");
18     }
19     else if (x > y)
20     {
21         printf("x é maior do que y\n");
22     }
23     else
24     {
25         printf("x é igual a y\n");
26     }
27 }
```

```
1  // Aritmética usando números de ponto flutuante do tipo double
2
3  #include <cs50.h>
4  #include <stdio.h>
5
6  int main(void)
7  {
8      // Pede ao usuário para informar x
9      double x = get_double("x: ");
10
11     // Pede ao usuário para informar y
12     double y = get_double("y: ");
13
14     // Calcula a divisão
15     printf("x / y = %.50f\n", x / y);
16 }
```

```
1 // get_float e printf com %f
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     int preco = get_float("Qual é o preço?\n$");
9     printf("O valor total é $%.2f.\n", preco * 1.0625);
10 }
```



```
1 // Aritmética usando números de ponto-flutuante (float)
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Pedir ao usuário para informar x
9     float x = get_float("x: ");
10
11     // Pedir ao usuário para informar y
12     float y = get_float("y: ");
13
14     // Calcular a divisão
15     printf("x / y = %.50f\n", x / y);
16 }
```

```
1  // get_int e printf com %i
2
3  #include <cs50.h>
4  #include <stdio.h>
5
6  int main(void)
7  {
8      int idade = get_int("Qual é sua idade?\n");
9      printf("Você tem pelo menos %i dias de idade.\n", idade * 365);
10 }
```

```
1 // Imprime uma linha com 4 pontos de interrogação
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     printf("????\n");
8 }
```

```
1 // Imprime uma linha com 4 pontos de interrogação usando uma repetição
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     for (int i = 0; i < 4; i++)
8     {
9         printf("?");
10    }
11    printf("\n");
12 }
```

```
1 // Imprime uma linha com um número n de pontos de interrogação usando uma repetição
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     int n;
9     do
10     {
11         n = get_int("Largura: ");
12     }
13     while (n < 1);
14
15     for (int i = 0; i < n; i++)
16     {
17         printf("?");
18     }
19     printf("\n");
20 }
```

```
1 // Imprime uma coluna com 3 blocos
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     printf("#\n");
8     printf("#\n");
9     printf("#\n");
10 }
```

```
1  // Imprime uma coluna com 3 blocos usando uma repetição
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7      for (int i = 0; i < 3; i++)
8      {
9          printf("#\n");
10     }
11 }
```

```
1 // Imprime uma coluna com um número n de blocos usando uma repetição
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     int n;
9     do
10     {
11         n = get_int("Altura: ");
12     }
13     while (n < 1);
14
15     for (int i = 0; i < n; i++)
16     {
17         printf("#\n");
18     }
19 }
```

```
1  // Imprime uma grade de blocos de tamanho 3 por 3
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7      printf("###\n");
8      printf("###\n");
9      printf("###\n");
10 }
```

```
1 // Imprime uma grade de blocos de tamanho 3 por 3 usando uma repetição
2
3 #include <stdio.h>
4
5 int main(void)
6 {
7     for (int i = 0; i < 3; i++)
8     {
9         printf("###\n");
10    }
11 }
```

```
1  // Imprime uma grade de blocos de tamanho n por n usando uma repetição
2
3  #include <cs50.h>
4  #include <stdio.h>
5
6  int main(void)
7  {
8      int n;
9      do
10     {
11         n = get_int("Tamanho: ");
12     }
13     while (n < 1);
14
15     for (int i = 0; i < n; i++)
16     {
17         for (int j = 0; j < n; j++)
18         {
19             printf("#");
20         }
21         printf("\n");
22     }
23 }
```

```
1  // Um programa que diz olá para o mundo
2
3  #include <stdio.h>
4
5  int main(void)
6  {
7      printf("olá, mundo\n");
8  }
```

```
1 // Calcula o resto da divisão
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     // Pede ao usuário que informe um inteiro
9     int n = get_int("n: ");
10
11     // Checa se o inteiro informado é par
12     if (n % 2 == 0)
13     {
14         printf("é par\n");
15     }
16     else
17     {
18         printf("é ímpar\n");
19     }
20 }
```

```
1  // Abstração e escopo
2
3  #include <cs50.h>
4  #include <stdio.h>
5
6  int receber_inteiro_positivo(void);
7
8  int main(void)
9  {
10     int i = receber_inteiro_positivo();
11     printf("%i\n", i);
12 }
13
14 // Pede ao usuário para informar um inteiro positivo
15 int receber_inteiro_positivo(void)
16 {
17     int n;
18     do
19     {
20         n = get_int("Inteiro positivo: ");
21     }
22     while (n < 1);
23     return n;
24 }
```

```
1  // Biblioteca de matemática
2
3  #include <cs50.h>
4  #include <math.h>
5  #include <stdio.h>
6
7  int main(void)
8  {
9      double base = get_double("Base: ");
10     double expoente = get_double("Expoente: ");
11     printf("Resultado: %.0f\n", pow(base, expoente));
12 }
```

```
1  // Retornando um valor com o comando return
2
3  #include <stdio.h>
4
5  void raizquadrada(int n);
6
7  int main(void)
8  {
9      int numero = get_int("Número: ");
10     printf("Resultado: %i\n", raizquadrada(n));
11 }
12
13 // Raiz quadrada de n
14 int raizquadrada(int n)
15 {
16     return n * n;
17 }
```



```
1 // get_string e printf com %s
2
3 #include <cs50.h>
4 #include <stdio.h>
5
6 int main(void)
7 {
8     string s = get_string("Qual é seu nome?\n");
9     printf("olá, %s\n", s);
10 }
```

```
1  // Transbordamento de inteiro
2
3  #include <stdio.h>
4  #include <unistd.h>
5
6  int main(void)
7  {
8      // Duplica o valor de i iterativamente
9      for (int i = 1; ; i *= 2)
10     {
11         printf("%i\n", i);
12         sleep(1);
13     }
14 }
```