



0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9

Utilize caneta azul ou preta e preencha completamente a quadrícula.

Exemplo: ~~☒~~ Não use ☒

Turma: (somente um número; consulte a pessoa responsável se não souber)

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------	--------------------------

← Marque as quadrículas ao lado para formar o seu número USP e escreva seu nome completo em letra legível na linha pontilhada abaixo. **Se seu número possui menos que 8 dígitos complete com zeros à esquerda.**

Nome: _____

.....

Esta prova tem duração de 120 minutos. Não desmonte a prova.

Q1 [2 pontos] Simule a execução do código abaixo e selecione a opção correspondente à saída impressa do programa. Ignore as quebras de linhas geradas pelo comando `cat(..., "\n")`.

```
f <- function(L,S) {
  l <- length(L)
  k <- 1
  while (k <= l) {
    r <- S[1]
    S <- S[2:length(S)]
    i <- 1
    while (i <= length(L)) {
      j <- 1 + (i + r) %% length(L)
      tmp <- L[i]
      L[i] <- L[j]
      L[j] <- tmp
      i <- i + 1
    }
    u <- L[length(L)]
    L <- L[1:length(L) - 1]
    cat(u, "\n")
    k <- k + 1
  }
}

main <- function() {
  f(c(31, 14, 119, 111, 106),c(3,5,7,11,13,17,19))
}

main()
```

Rascunho

- | | | |
|--|--|---|
| <input type="checkbox"/> 111 31 14 106 119 | <input type="checkbox"/> 106 14 31 119 111 | <input type="checkbox"/> 31 111 14 106 119 |
| <input type="checkbox"/> 14 31 111 106 119 | <input type="checkbox"/> 119 14 31 111 106 | <input type="checkbox"/> 31 111 14 119 106 |
| <input type="checkbox"/> 31 111 119 106 14 | <input type="checkbox"/> 106 111 14 31 119 | <input type="checkbox"/> 106 119 111 31 14 |
| <input type="checkbox"/> 119 31 106 14 111 | <input type="checkbox"/> 111 14 106 31 119 | <input checked="" type="checkbox"/> 106 31 111 14 119 |
| <input type="checkbox"/> 106 31 119 14 111 | <input type="checkbox"/> 111 14 119 106 31 | <input type="checkbox"/> 31 14 111 106 119 |



Q2 [3 pontos] Cartões de crédito geralmente têm 16 dígitos. Os primeiros dígitos à esquerda indicam a bandeira do cartão (Visa começa com dígito 4, Mastercard varia de 51 a 55). O último dígito (mais à direita) é um dígito verificador calculado a partir dos demais dígitos pelo algoritmo de Luhn (explicado no quadro abaixo) como medida de segurança, a fim de evitar falsificações e erros de digitação.

Preencha as lacunas no código abaixo de forma a obter um programa que, dado um inteiro com os 16 dígitos de um cartão, checa se seu dígito verificador está correto. Em caso afirmativo, o programa ainda informa se a bandeira é Mastercard.

```
cartão <- as.numeric(readline("Número do cartão: "))
dv <- cartão %% 10
falta <- cartão %/% 10
par <- TRUE
soma <- 0
L1
  atual <- falta %% 10
  if (par) {
    val <- 2 * atual
    L2
  } else {
    val <- atual
  }
  soma <- soma + val
  L3
  L4
}
L5
if (dv == calculado) {
  L6
  L7
  L8
}
cat("cartão válido", mastercard, "\n")
} else {
  cat("cartão inválido: dígito deveria ser", calculado, "\n")
}
```

Algoritmo de Luhn

Exemplo: vamos conferir se o dígito verificador 7 está correto no cartão de crédito número 4984232050486427.

Numerando os dígitos da direita para esquerda, iniciando em 1, tomamos os dígitos em posições pares (grifadas) multiplicados por dois. Caso o produto resulte em algum número de 2 dígitos, estes devem ser somados. Logo, para o exemplo temos:

$$2 \times 2, 6 \times 2, 4 \times 2, 5 \times 2, 2 \times 2, 2 \times 2, 8 \times 2, 4 \times 2 \Rightarrow 4, 12, 8, 10, 4, 4, 16, 8 \Rightarrow 4, 3, 8, 1, 4, 4, 7, 8$$

Calculamos então a soma dos valores obtidos:

$$4 + 3 + 8 + 1 + 4 + 4 + 7 + 8 = 39$$

A seguir, adicionamos ao somatório todos os dígitos em posições ímpares que não foram usados na etapa anterior, com exceção do primeiro à direita. Assim, temos:

$$39 + 9 + 4 + 3 + 0 + 0 + 8 + 4 = 39 + 28 = 67$$

Para 67 ser um múltiplo de 10, faltam 3 unidades, então 3 é o nosso dígito verificador e, portanto, o valor 7 à direita está incorreto; o cartão 4984232050486427 é inválido.



Preencha as lacunas indicadas assinalando as respostas correspondentes abaixo (apenas uma para cada linha). Não tente montar o programa testando as combinações possíveis, pois não vai dar tempo; procure primeiro identificar as principais variáveis do programa, seus significados e a estrutura do algoritmo usado.

L1:	<input type="checkbox"/> while (falta %% 10 != 0) { <input type="checkbox"/> while (falta >= 0) {	<input type="checkbox"/> while (falta %% 10 > 0) { <input checked="" type="checkbox"/> while (falta > 0) {	<input type="checkbox"/> while (par) {
L2:	<input checked="" type="checkbox"/> val <- val %% 10 + val %% 10 <input type="checkbox"/> val <- val %% 10 + val	<input type="checkbox"/> val <- val + 1 <input type="checkbox"/> val <- val %% 10 + 2 * atual	<input type="checkbox"/> val <- val + (val %% 10)
L3:	<input checked="" type="checkbox"/> falta <- falta %% 10 <input type="checkbox"/> falta <- falta - 1	<input type="checkbox"/> falta <- 2 * falta %% 10 <input type="checkbox"/> falta <- falta - soma	<input type="checkbox"/> falta <- falta %% 10
L4:	<input type="checkbox"/> par <- val > 0 <input type="checkbox"/> par <- falta > 0	<input checked="" type="checkbox"/> par <- ! par <input type="checkbox"/> par <- FALSE par	<input type="checkbox"/> par <- TRUE par
L5:	<input checked="" type="checkbox"/> calculado <- 10 - (soma %% 10) <input type="checkbox"/> calculado <- soma %% 10	<input type="checkbox"/> calculado <- (soma + 10) %% 10 <input type="checkbox"/> calculado <- soma %% 10**15	<input type="checkbox"/> calculado <- (10 - soma) %% 10
L6:	<input type="checkbox"/> mastercard <- "(Mastercard)" <input checked="" type="checkbox"/> mastercard = ""	<input type="checkbox"/> mastercard <- TRUE <input type="checkbox"/> mastercard <- FALSE	<input type="checkbox"/> mastercard <- 0
L7:	<input checked="" type="checkbox"/> if (cartão %% 10**14 >= 51 && cartão %% 10**14 <= 55) { <input type="checkbox"/> if (not (cartão %% 10**14 == 4)) { <input type="checkbox"/> if (cartão %% 10**14 == 51 cartão %% 10**14 == 55) { <input type="checkbox"/> if (cartão %% 10**14 > 51 && cartão %% 10**14 < 55) { <input type="checkbox"/> if (not (cartão %% 10**14 <= 51 cartão %% 10**14 >= 55)) {		
L8:	<input type="checkbox"/> mastercard = 0 <input type="checkbox"/> mastercard = TRUE	<input type="checkbox"/> mastercard = FALSE <input type="checkbox"/> mastercard = ""	<input checked="" type="checkbox"/> mastercard <- "(Mastercard)"



Q3 [3 pontos] Considere as seguintes definições:

Números amigos: dizemos que dois naturais a e b são amigos se cada um deles é igual à soma dos divisores próprios do outro. Um exemplo é o par 220 e 284, uma vez que os divisores próprios de 220 são 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 e 110, que somam 284, e os divisores próprios de 284 são 1, 2, 4, 71 e 142, que somam 220.

Números mutuamente amigos: dois números a e b são mutuamente amigos se a razão entre a soma de todos os divisores do número a e o número a é igual à razão entre a soma de todos os divisores do número b e o número b . Por exemplo, os números 30 e 140 são mutuamente amigos, pois:

- para 30 temos $(1 + 2 + 3 + 5 + 6 + 10 + 15 + 30)/30 = 72/30 = 12/5$;
- para 140 temos $(1 + 2 + 4 + 5 + 7 + 10 + 14 + 20 + 28 + 35 + 70 + 140)/140 = 336/140 = 12/5$.

Números primos entre si: dois inteiros a e b são primos entre si se o único divisor comum a ambos é 1.

Preencha as lacunas no código abaixo de forma a obter um programa que, dados dois inteiros positivos distintos a e b ($a > 0$, $b > 0$ e $a \neq b$), verifica se eles satisfazem as definições anteriores. Não tente montar o programa testando as combinações possíveis, pois não vai dar tempo; procure primeiro identificar as principais variáveis do programa, seus significados e a estrutura do algoritmo usado.

```
a <- as.integer(readline("Digite a: "))
b <- as.integer(readline("Digite b: "))
i <- 1
sa <- 0
L1
    L2
        L3
    }
    i <- i + 1
}
cat(a, "gera soma dos divisores próprios", sa, "\n")
i <- 1
sb <- 0
L4
    L5
        L6
    }
    i <- i + 1
}
cat(b, "gera soma dos divisores próprios", sb, "\n")
mdc <- a
L7
    mdc <- mdc - 1
}
L8
    cat("primos entre si", "\n")
}
if (sa == b && sb == a) {
    cat("números amigos", "\n")
}
L9
    cat("números mutuamente amigos", "\n")
}
```



L1:	<input checked="" type="checkbox"/> while (i < a) { <input type="checkbox"/> while (i <= a) {	<input type="checkbox"/> while (sa < a) { <input type="checkbox"/> while (i <= sa) {	<input type="checkbox"/> while (i > 0) {
L2:	<input type="checkbox"/> if (sa == i) { <input type="checkbox"/> if (i + sa < a) {	<input checked="" type="checkbox"/> if (a %% i == 0) { <input type="checkbox"/> if (a %% i != 0) {	<input type="checkbox"/> if (a %% i > 0) {
L3:	<input type="checkbox"/> sa <- i <input checked="" type="checkbox"/> sa <- sa + i	<input type="checkbox"/> sa <- i * a <input type="checkbox"/> sa <- sa + 1	<input type="checkbox"/> sa <- sa + i * a
L4:	<input type="checkbox"/> while (i <= b) { <input type="checkbox"/> while (sb < b) {	<input checked="" type="checkbox"/> while (i < b) { <input type="checkbox"/> while (i <= sb) {	<input type="checkbox"/> while (i > 0) {
L5:	<input type="checkbox"/> if (b %% i > 0) { <input type="checkbox"/> if (b %% i != 0) {	<input checked="" type="checkbox"/> if (b %% i == 0) { <input type="checkbox"/> if (sb == i) {	<input type="checkbox"/> if (i + sb < b) {
L6:	<input type="checkbox"/> sb <- i * b <input type="checkbox"/> sb <- sb + 1	<input type="checkbox"/> sb <- i <input checked="" type="checkbox"/> sb <- sb + i	<input type="checkbox"/> sb <- sb + i * b
L7:	<input type="checkbox"/> while (a %% mdc == 0 && b %% mdc != 0) { <input checked="" type="checkbox"/> while (a %% mdc != 0 b %% mdc != 0) { <input type="checkbox"/> while (a %% mdc == 0 b %% mdc != 0) {	<input type="checkbox"/> while (a %% mdc != 0 b %% mdc == 0) { <input type="checkbox"/> while (a %% mdc != 0 && b %% mdc != 0) {	
L8:	<input type="checkbox"/> if (mdc > 1) { <input type="checkbox"/> if (mdc > 0) {	<input type="checkbox"/> if (mdc >= 0) { <input type="checkbox"/> if (mdc >= 1) {	<input checked="" type="checkbox"/> if (mdc == 1) {
L9:	<input type="checkbox"/> if ((sa + a) / a / (sb + b) / b == 1) { <input type="checkbox"/> if ((sa + a) * a == (sb + b) * b) { <input type="checkbox"/> if ((sa + a) // a == (sb + b) // b) {	<input checked="" type="checkbox"/> if ((sa + a) * b == (sb + b) * a) { <input type="checkbox"/> if ((sa + a) // b == (sb + b) // a) {	



Q4 [2 pontos] Considere o seguinte programa e analise as afirmações abaixo.

```
main <- function() {  
  x <- as.integer(readline("Digite x: "))  
  
  if (x < 0) {  
    x <- -x  
  }  
  
  if (x > 10) {  
    cat("muito", "\n")  
  } else {  
    if (x > 5) {  
      cat("bom", "\n")  
    }  
  }  
}  
  
main()
```

I - O programa não exibe nenhuma mensagem se o usuário digitar o valor 2.

II - O programa exibe a mensagem muito bom se o usuário digitar o valor 15 .

III - O programa exibe a mensagem bom se o usuário digitar o valor -7

Apenas I e II são corretas.

Apenas III é correta.

Apenas II e III são corretas.

Apenas I é correta.

Todas são falsas.

Todas são corretas.

Apenas II é correta.

Apenas I e III são corretas.