MC102

Algoritmos e Programação de Computadores

Prova 1

Turmas A B C D E F I J K L

Segundo Semestre de 2019

Questão	Nota
1.a	
1.b	
1.c	
2	
3	
Total	

Nome: RA:

Importante: Esta avaliação é individual. Não é permitida a consulta a qualquer material impresso, nem o uso de celulares ou outros dispositivos eletrônicos de comunicação/computação. As respostas deverão ser escritas nos espaços demarcados, opcionalmente a lápis. Não se esqueça de indentar corretamente os códigos solicitados. **Boa prova!**

1. Em cada série, para cada trecho de código, indique o que será escrito quando os programas forem executados. Caso um programa execute corretamente e não produza nenhuma saída, preencha com "__" o espaço da resposta correspondente. Caso algum erro seja encontrado, indique o motivo e marque no código o ponto em que ele ocorre.

a) (1.2 ponto) **Tipos básicos e precedência de operadores**

```
# int ou float?
v = 2.0 + 7
print(v)
```

9.0

```
q = 8 // 3
r = 5 % 2
print(q, r)
```

2 1

```
a = 5 * 3 - 1
b = 3 - 1 / 2 + 3
print(a, b)
```

14 5.5

Dicas: // executa divisão inteira e % obtém o resto da divisão

---

```
# True ou
# False?
print(10 > 2)
```

True

```
a = 3.0 + 1
a = a > 2
print(a)
```

True

```
a = True
b = False
print(not a, a or b and b)
```

False True

```
s1 = "O"
s2 = "+"
print(s1 + s2)
```

O+

```
s1 = "O+"
s2 = "A+"
print("s1 != s2")
```

s1 != s2

```
a = 5 - 3 < 2 + 2
b = "resultado ="
print(b, a)
```

resultado = True

---

b) (0.8 ponto) **Comandos condicionais**

```
a = 7
b = 3
if a > b:
  print(a + b)
if b > a:
  print(b - a)
else:
  print(0)
```

```
10
0
```

```
a = 2
b = 2
if a > b:
  print(a - b)
if b >= a:
  print(b - a)
if a == b:
  print(0)
```

```
0
0
```

```
a = 5
b = 5
if a > b:
  print(a - b)
else:
  print(b - a)
elif a == b:
  print(0)
```

```
invalid syntax
(string, line 7)
```

```
a = 7
b = 8
if a > b:
  print(a - b)
elif b > a:
  print(b - a)
else:
  print(0)
```

```
1
```

---

c) (2.5 pontos) **Listas e comandos repetitivos**

```
lista = []
i = 0
while i < 10 :
  lista.append(i)
  i = i + 2
print(lista)
```

```
[0, 2, 4, 6, 8]
```

```
lista = [1, 3, 4, 5, 12]
for i in range(len(lista)) :
  if lista[i] % 2 == 1 :
    lista[i] = lista[i] + 1
print(lista)
```

```
[2, 4, 4, 6, 12]
```

```
lista_a = [2, 1, 3, 7, 4]
lista_b = [0, 5, 2, 1, 6]
lista_c = []
for i in range(len(lista_a)) :
  if lista_a[i] > lista_b[i] :
     lista_c.append(lista_a[i])
  else:
     lista_c.append(lista_b[i])
print(lista_c)
```

```
[2, 5, 3, 7, 6]
```

Dicas: A função `len(lista)` retorna o número de elementos do objeto `lista` que foi passado como parâmetro.

O método `lista.append(elem)` adiciona o elemento `elem` ao final de um objeto `lista`.

Veja exemplos do uso da função `range()` na última página desta prova.

```
lista_tuplas = [(0,1), (2,1), (1,2), (3,1)]
lista = []
total = 0
for i in range(len(lista_tuplas)) :
    lista.append(lista_tuplas[i][0] * lista_tuplas[i][1])
    total = total + lista[i]
print(lista)
print(total)
```

```
[0, 2, 2, 3]
7
```

---

2. (2.5 pontos) Como vimos na Tarefa de Laboratório 4, para ser doador de sangue é necessário atender a uma série de requisitos. Observe abaixo um pequeno programa que lê um dado sobre peso corporal e escreve uma mensagem caso a pessoa não atinja o peso mínimo de 50.0 kg.

```
peso = float(input())
if peso < 50.0 :
    print("Impedimento: abaixo do peso mínimo.")
```

Baseando-se no modelo acima, escreva um programa que lê um inteiro que representa os anos completos da idade de uma pessoa e escreve, conforme o caso, uma das mensagens abaixo. O uso da estrutura condicional **if**, **elif** e **else** é obrigatório.

- idade $<$ 16: `"Impedimento: menor de 16 anos."`
- idade $>$ 69: `"Impedimento: maior de 69 anos."`
- 16 $\leq$ idade $<$ 18: `"Restrição: requer autorização do responsável."`
- 60 $\leq$ idade $\leq$ 69: `"Restrição: não pode ser a primeira doação."`
- 18 $\leq$ idade $<$ 60: `"Sem impedimentos ou restrições."`

```
idade = int(input())
if idade < 16 :
   print("Impedimento: menor de 16 anos.")
elif idade < 18 :
   print("Restrição: requer autorização do responsável.")
elif idade < 60 :
   print("Sem impedimentos ou restrições.")
elif idade <= 69 :
   print("Restrição: não pode ser a primeira doação.")
else :
    print("Impedimento: maior de 69 anos.")
```

3. (3.0 pontos) Observe os dados de entrada e as saídas abaixo de maneira a identificar padrões. Em seguida, escreva um programa completo que lê um inteiro da entrada e produz a saída correspondente ao padrão apresentado. Seu código deve estar estruturado utilizando o comando **for** e a função `range()` e pode considerar que o inteiro fornecido está no intervalo [3..9].

Entrada: 3
0123
012
01
0
3210

Entrada: 4
01234
0123
012
01
0
43210

Entrada: 5
012345
01234
0123
012
01
0
543210

```
x = int(input())
for i in range(x, -1, -1) :
  for j in range(i+1) :
    print(j, end="")
  print()
for j in range(x,-1,-1) :
   print(j, end="")
print()
```

Dicas: A função `print(v, end="")` escreve o valor da variável `v` sem pular linha ao final.

A função `range()` gera uma sequência de números útil para iteração em laços do tipo `for`.
Uso: `range(fim)`, `range(início, fim)` ou `range(início, fim, passo)`. Observe os exemplos e as saídas correspondentes:

```
for i in range(2) :
  print(i, end="")

01
```

```
for i in range(1,3) :
  print(i, end="")

12
```

```
for i in range(3,1,-1) :
  print(i, end="")

32
```

MC102

Algoritmos e Programação de Computadores

Prova 1

Turmas A B C D E F I J K L

Segundo Semestre de 2019

Questão	Nota
1.a	
1.b	
1.c	
2	
3	
Total	

Nome: RA:

Importante: Esta avaliação é individual. Não é permitida a consulta a qualquer material impresso, nem o uso de celulares ou outros dispositivos eletrônicos de comunicação/computação. As respostas deverão ser escritas nos espaços demarcados, opcionalmente a lápis. Não se esqueça de indentar corretamente os códigos solicitados. **Boa prova!**

1. Em cada série, para cada trecho de código, indique o que será escrito quando os programas forem executados. Caso um programa execute corretamente e não produza nenhuma saída, preencha com "__" o espaço da resposta correspondente. Caso algum erro seja encontrado, indique o motivo e marque no código o ponto em que ele ocorre.

a) (1.2 ponto) **Tipos básicos e precedência de operadores**

```
# int ou float?
v = 4.0 * 2
print(v)
```

8.0

```
q = 11 // 3
r = 5 % 3
print(q, r)
```

3 2

```
a = 4 * 3 - 1
b = 2 - 1 / 2 + 3
print(a, b)
```

11 4.5

Dicas: // executa divisão inteira e % obtém o resto da divisão

---

```
# True ou
# False?
print(10 < 2)
```

False

```
a = 3.0 + 1
a = a < 2
print(a)
```

False

```
a = True
b = False
print(not b, b or a and b)
```

True False

```
s1 = "AB"
s2 = "+"
print(s1 + s2)
```

AB+

```
s1 = "A+"
s2 = "B-"
print("s1 == s2")
```

s1 == s2

```
a = 5 - 3 > 2 + 2
b = "resultado ="
print(b, a)
```

resultado = False

---

b) (0.8 ponto) **Comandos condicionais**

```
a = 6
b = 3
if a > b:
  print(a + b)
if b > a:
  print(b - a)
else:
  print(0)
```

9
0

```
a = 5
b = 5
if a > b:
  print(a - b)
else:
  print(b - a)
elif a == b:
  print(0)
```

invalid syntax (string, line 7)

```
a = 2
b = 2
if a > b:
  print(a - b)
if b >= a:
  print(b - a)
if a == b:
  print(0)
```

0
0

```
a = 7
b = 3
if a > b:
  print(a - b)
elif b > a:
  print(b - a)
else:
  print(0)
```

4

---

c) (2.5 pontos) **Listas e comandos repetitivos**

```
lista = []
i = 1
while i < 11 :
  lista.append(i)
  i = i + 2
print(lista)
```

[1, 3, 5, 7, 9]

```
lista = [2, 3, 4, 7, 12]
for i in range(len(lista)) :
  if lista[i] % 2 == 0 :
    lista[i] = lista[i] + 1
print(lista)
```

[3, 3, 5, 7, 13]

```
lista_a = [2, 1, 3, 7, 4]
lista_b = [0, 5, 2, 1, 6]
lista_c = []
for i in range(len(lista_a)) :
  if lista_a[i] < lista_b[i] :
     lista_c.append(lista_a[i])
  else:
     lista_c.append(lista_b[i])
print(lista_c)
```

[0, 1, 2, 1, 4]

Dicas: A função `len(lista)` retorna o número de elementos do objeto `lista` que foi passado como parâmetro.

O método `lista.append(elem)` adiciona o elemento `elem` ao final de um objeto `lista`.

Veja exemplos do uso da função `range()` na última página desta prova.

```
lista_tuplas = [(0,1), (2,1), (1,2), (3,3)]
lista = []
total = 0
for i in range(len(lista_tuplas)) :
    lista.append(lista_tuplas[i][0] * lista_tuplas[i][1])
    total = total + lista[i]
print(lista)
print(total)
```

[0, 2, 2, 9]
13

---

2. (2.5 pontos) Como vimos na Tarefa de Laboratório 4, para ser doador de sangue é necessário atender a uma série de requisitos. Observe abaixo um pequeno programa que lê um dado sobre peso corporal e escreve uma mensagem caso a pessoa não atinja o peso mínimo de 50.0 kg.

```
peso = float(input())
if peso < 50.0 :
    print("Impedimento: abaixo do peso mínimo.")
```

Baseando-se no modelo acima, escreva um programa que lê um inteiro que representa os anos completos da idade de uma pessoa e escreve, conforme o caso, uma das mensagens abaixo. O uso da estrutura condicional **if**, **elif** e **else** é obrigatório.

- idade $<$ 16: `"Impedimento: menor de 16 anos."`
- idade $>$ 69: `"Impedimento: maior de 69 anos."`
- 16 $\leq$ idade $<$ 18: `"Restrição: requer autorização do responsável."`
- 60 $\leq$ idade $\leq$ 69: `"Restrição: não pode ser a primeira doação."`
- 18 $\leq$ idade $<$ 60: `"Sem impedimentos ou restrições."`

```
idade = int(input())
if idade < 16 :
   print("Impedimento: menor de 16 anos.")
elif idade < 18 :
   print("Restrição: requer autorização do responsável.")
elif idade < 60 :
   print("Sem impedimentos ou restrições.")
elif idade <= 69 :
   print("Restrição: não pode ser a primeira doação.")
else :
    print("Impedimento: maior de 69 anos.")
```

3. (3.0 pontos) Observe os dados de entrada e as saídas abaixo de maneira a identificar padrões. Em seguida, escreva um programa completo que lê um inteiro da entrada e produz a saída correspondente ao padrão apresentado. Seu código deve estar estruturado utilizando o comando **for** e a função `range()` e pode considerar que o inteiro fornecido está no intervalo [3..9].

Entrada: 3
3210
0123
012
01
0

Entrada: 4
43210
01234
0123
012
01
0

Entrada: 5
543210
012345
01234
0123
012
01
0

```
x = int(input())
for j in range(x,-1,-1) :
   print(j, end="")
print()
for i in range(x, -1, -1) :
  for j in range(i+1) :
    print(j, end="")
  print()
```

Dicas: A função `print(v, end="")` escreve o valor da variável `v` sem pular linha ao final.

A função `range()` gera uma sequência de números útil para iteração em laços do tipo `for`.
Uso: `range(fim)`, `range(início, fim)` ou `range(início, fim, passo)`. Observe os exemplos e as saídas correspondentes:

```
for i in range(2) :
  print(i, end="")

01
```

```
for i in range(1,3) :
  print(i, end="")

12
```

```
for i in range(3,1,-1) :
  print(i, end="")

32
```

MC102

Algoritmos e Programação de Computadores

Prova 1

Turmas A B C D E F I J K L

Segundo Semestre de 2019

Questão	Nota
1.a	
1.b	
1.c	
2	
3	
Total	

Nome: RA:

Importante: Esta avaliação é individual. Não é permitida a consulta a qualquer material impresso, nem o uso de celulares ou outros dispositivos eletrônicos de comunicação/computação. As respostas deverão ser escritas nos espaços demarcados, opcionalmente a lápis. Não se esqueça de indentar corretamente os códigos solicitados. **Boa prova!**

1. Em cada série, para cada trecho de código, indique o que será escrito quando os programas forem executados. Caso um programa execute corretamente e não produza nenhuma saída, preencha com "__" o espaço da resposta correspondente. Caso algum erro seja encontrado, indique o motivo e marque no código o ponto em que ele ocorre.

a) (1.2 ponto) **Tipos básicos e precedência de operadores**

```
# int ou float?
v = 9.0 - 2
print(v)
```

7.0

```
q = 14 // 3
r = 6 % 2
print(q, r)
```

4 0

```
a = 3 * 3 - 1
b = 3 - 1 / 4 + 2
print(a, b)
```

8 4.75

Dicas: // executa divisão inteira e % obtém o resto da divisão

---

```
# True ou
# False?
print(7 != 2)
```

True

```
a = 2.5 + 1
a = a > 5
print(a)
```

False

```
a = True
b = False
print(not a, b or a and b)
```

False False

```
s1 = "B"
s2 = "+"
print(s1 + s2)
```

B+

```
s1 = "0+"
s2 = "0-"
print("s1 != s2")
```

s1 != s2

```
a = 5 - 3 == 1 + 1
b = "resultado ="
print(b, a)
```

resultado = True

---

b) (0.8 ponto) **Comandos condicionais**

```
a = 2
b = 2
if a > b:
  print(a - b)
if b >= a:
  print(b - a)
if a == b:
  print(0)
```

0
0

```
a = 6
b = 2
if a > b:
  print(a + b)
if b > a:
  print(b - a)
else:
  print(0)
```

8
0

```
a = 7
b = 9
if a > b:
  print(a - b)
elif b > a:
  print(b - a)
else:
  print(0)
```

2

```
a = 5
b = 5
if a > b:
  print(a - b)
else:
  print(b - a)
elif a == b:
  print(0)
```

invalid syntax (string, line 7)

---

c) (2.5 pontos) **Listas e comandos repetitivos**

```
lista = []
i = 0
while i < 15 :
  lista.append(i)
  i = i + 3
print(lista)
```

[0, 3, 6, 9, 12]

```
lista = [0, 1, 3, 6, 11]
for i in range(len(lista)) :
  if lista[i] % 2 == 1 :
    lista[i] = lista[i] + 1
print(lista)
```

[0, 2, 4, 6, 12]

```
lista_a = [1, 3, 7, 8, 5]
lista_b = [0, 5, 2, 1, 6]
lista_c = []
for i in range(len(lista_a)) :
  if lista_a[i] > lista_b[i] :
     lista_c.append(lista_a[i])
  else:
     lista_c.append(lista_b[i])
print(lista_c)
```

[1, 5, 7, 8, 6]

Dicas: A função `len(lista)` retorna o número de elementos do objeto `lista` que foi passado como parâmetro.

O método `lista.append(elem)` adiciona o elemento `elem` ao final de um objeto `lista`.

Veja exemplos do uso da função `range()` na última página desta prova.

```
lista_tuplas = [(1,1), (3,1), (1,2), (3,1)]
lista = []
total = 0
for i in range(len(lista_tuplas)) :
    lista.append(lista_tuplas[i][0] * lista_tuplas[i][1])
    total = total + lista[i]
print(lista)
print(total)
```

[1, 3, 2, 3]
9

---

2. (2.5 pontos) Como vimos na Tarefa de Laboratório 4, para ser doador de sangue é necessário atender a uma série de requisitos. Observe abaixo um pequeno programa que lê um dado sobre peso corporal e escreve uma mensagem caso a pessoa não atinja o peso mínimo de 50.0 kg.

```
peso = float(input())
if peso < 50.0 :
    print("Impedimento: abaixo do peso mínimo.")
```

Baseando-se no modelo acima, escreva um programa que lê um inteiro que representa os anos completos da idade de uma pessoa e escreve, conforme o caso, uma das mensagens abaixo. O uso da estrutura condicional **if**, **elif** e **else** é obrigatório.

- idade $<$ 16: `"Impedimento: menor de 16 anos."`
- idade $>$ 69: `"Impedimento: maior de 69 anos."`
- 16 $\leq$ idade $<$ 18: `"Restrição: requer autorização do responsável."`
- 60 $\leq$ idade $\leq$ 69: `"Restrição: não pode ser a primeira doação."`
- 18 $\leq$ idade $<$ 60: `"Sem impedimentos ou restrições."`

```
idade = int(input())
if idade < 16 :
   print("Impedimento: menor de 16 anos.")
elif idade < 18 :
   print("Restrição: requer autorização do responsável.")
elif idade < 60 :
   print("Sem impedimentos ou restrições.")
elif idade <= 69 :
   print("Restrição: não pode ser a primeira doação.")
else :
    print("Impedimento: maior de 69 anos.")
```

3. (3.0 pontos) Observe os dados de entrada e as saídas abaixo de maneira a identificar padrões. Em seguida, escreva um programa completo que lê um inteiro da entrada e produz a saída correspondente ao padrão apresentado. Seu código deve estar estruturado utilizando o comando **for** e a função `range()` e pode considerar que o inteiro fornecido está no intervalo [3..9].

Entrada: 3
3210
0
01
012
0123

Entrada: 4
43210
0
01
012
0123
01234

Entrada: 5
543210
0
01
012
0123
01234
012345

```
x = int(input())
for j in range(x,-1,-1) :
   print(j, end="")
print()
for i in range(x+1) :
  for j in range(i+1) :
    print(j, end="")
  print()
```

Dicas: A função `print(v, end="")` escreve o valor da variável `v` sem pular linha ao final.

A função `range()` gera uma sequência de números útil para iteração em laços do tipo `for`.
Uso: `range(fim)`, `range(início, fim)` ou `range(início, fim, passo)`. Observe os exemplos e as saídas correspondentes:

```
for i in range(2) :
  print(i, end="")

01
```

```
for i in range(1,3) :
  print(i, end="")

12
```

```
for i in range(3,1,-1) :
  print(i, end="")

32
```

MC102

Algoritmos e Programação de Computadores

Prova 1

Turmas A B C D E F I J K L

Segundo Semestre de 2019

Questão	Nota
1.a	
1.b	
1.c	
2	
3	
Total	

Nome: RA:

Importante: Esta avaliação é individual. Não é permitida a consulta a qualquer material impresso, nem o uso de celulares ou outros dispositivos eletrônicos de comunicação/computação. As respostas deverão ser escritas nos espaços demarcados, opcionalmente a lápis. Não se esqueça de indentar corretamente os códigos solicitados. **Boa prova!**

1. Em cada série, para cada trecho de código, indique o que será escrito quando os programas forem executados. Caso um programa execute corretamente e não produza nenhuma saída, preencha com "__" o espaço da resposta correspondente. Caso algum erro seja encontrado, indique o motivo e marque no código o ponto em que ele ocorre.

a) (1.2 ponto) **Tipos básicos e precedência de operadores**

```
# int ou float?
v = 3 - 1.0
print(v)
```

2.0

```
q = 5 // 3
r = 11 % 4
print(q, r)
```

1 3

```
a = 2 * 4 - 1
b = 1 - 1 / 4 + 1
print(a, b)
```

7 1.75

Dicas: // executa divisão inteira e % obtém o resto da divisão

---

```
# True ou
# False?
print(7 == 2)
```

False

```
a = 3.0 + 1
a = a > 2
print(a)
```

True

```
a = True
b = False
print(not b, a or b and b)
```

True True

```
s1 = "A"
s2 = "+"
print(s1 + s2)
```

A+

```
s1 = "AB+"
s2 = "B+"
print("s1 == s2")
```

s1 == s2

```
a = 5 - 3 != 1 + 1
b = "resultado ="
print(b, a)
```

resultado = False

---

b) (0.8 ponto) **Comandos condicionais**

```
a = 7
b = 3
if a > b:
  print(a - b)
else:
  print(b - a)
elif a == b:
  print(0)
```

invalid syntax (string, line 7)

```
a = 5
b = 1
if a > b:
  print(a + b)
if b > a:
  print(b - a)
else:
  print(0)
```

6
0

```
a = 10
b = 7
if a > b:
  print(a - b)
elif b > a:
  print(b - a)
else:
  print(0)
```

3

```
a = 4
b = 4
if a > b:
  print(a - b)
if b >= a:
  print(b - a)
if a == b:
  print(0)
```

0
0

---

c) (2.5 pontos) **Listas e comandos repetitivos**

```
lista = []
i = 0
while i < 20 :
  lista.append(i)
  i = i + 4
print(lista)
```

[0, 4, 8, 12, 16]

```
lista = [0, 2, 4, 5, 12]
for i in range(len(lista)) :
  if lista[i] % 2 == 0 :
    lista[i] = lista[i] + 1
print(lista)
```

[1, 3, 5, 5, 13]

```
lista_a = [1, 3, 7, 8, 5]
lista_b = [0, 5, 2, 1, 6]
lista_c = []
for i in range(len(lista_a)) :
  if lista_a[i] < lista_b[i] :
     lista_c.append(lista_a[i])
  else:
     lista_c.append(lista_b[i])
print(lista_c)
```

[0, 3, 2, 1, 5]

Dicas: A função `len(lista)` retorna o número de elementos do objeto `lista` que foi passado como parâmetro.

O método `lista.append(elem)` adiciona o elemento `elem` ao final de um objeto `lista`.

Veja exemplos do uso da função `range()` na última página desta prova.

```
lista_tuplas = [(0,1), (3,1), (2,2), (3,1)]
lista = []
total = 0
for i in range(len(lista_tuplas)) :
    lista.append(lista_tuplas[i][0] * lista_tuplas[i][1])
    total = total + lista[i]
print(lista)
print(total)
```

[0, 3, 4, 3]
10

---

2. (2.5 pontos) Como vimos na Tarefa de Laboratório 4, para ser doador de sangue é necessário atender a uma série de requisitos. Observe abaixo um pequeno programa que lê um dado sobre peso corporal e escreve uma mensagem caso a pessoa não atinja o peso mínimo de 50.0 kg.

```
peso = float(input())
if peso < 50.0 :
    print("Impedimento: abaixo do peso mínimo.")
```

Baseando-se no modelo acima, escreva um programa que lê um inteiro que representa os anos completos da idade de uma pessoa e escreve, conforme o caso, uma das mensagens abaixo. O uso da estrutura condicional **if**, **elif** e **else** é obrigatório.

- idade $<$ 16: `"Impedimento: menor de 16 anos."`
- idade $>$ 69: `"Impedimento: maior de 69 anos."`
- 16 $\leq$ idade $<$ 18: `"Restrição: requer autorização do responsável."`
- 60 $\leq$ idade $\leq$ 69: `"Restrição: não pode ser a primeira doação."`
- 18 $\leq$ idade $<$ 60: `"Sem impedimentos ou restrições."`

```
idade = int(input())
if idade < 16 :
   print("Impedimento: menor de 16 anos.")
elif idade < 18 :
   print("Restrição: requer autorização do responsável.")
elif idade < 60 :
   print("Sem impedimentos ou restrições.")
elif idade <= 69 :
   print("Restrição: não pode ser a primeira doação.")
else :
    print("Impedimento: maior de 69 anos.")
```

3. (3.0 pontos) Observe os dados de entrada e as saídas abaixo de maneira a identificar padrões. Em seguida, escreva um programa completo que lê um inteiro da entrada e produz a saída correspondente ao padrão apresentado. Seu código deve estar estruturado utilizando o comando **for** e a função `range()` e pode considerar que o inteiro fornecido está no intervalo [3..9].

Entrada: 3
0
01
012
0123
3210

Entrada: 4
0
01
012
0123
01234
43210

Entrada: 5
0
01
012
0123
01234
012345
543210

```
x = int(input())
for i in range(x+1) :
  for j in range(i+1) :
    print(j, end="")
  print()
for j in range(x,-1,-1) :
   print(j, end="")
print()
```

Dicas: A função `print(v, end="")` escreve o valor da variável `v` sem pular linha ao final.

A função `range()` gera uma sequência de números útil para iteração em laços do tipo `for`.
Uso: `range(fim)`, `range(início, fim)` ou `range(início, fim, passo)`. Observe os exemplos e as saídas correspondentes:

```
for i in range(2) :
  print(i, end="")

01
```

```
for i in range(1,3) :
  print(i, end="")

12
```

```
for i in range(3,1,-1) :
  print(i, end="")

32
```