A 10 POINTS

Full Name ____________

Section & Subsection ____________

Roll # ____________

1. (2 points) The following fragment of code was intended to be part of a telephone service provider (like BSNL). Initially, the exchange checks if the receiver of the call is busy, and if so must play a hold tone for the caller and retry the same process after playing it. The hold tone should be played for at most 5 minutes. If receiver is not busy, the caller should be connected to the receiver and the program terminates after disconnection.

   The `is_busy` function takes a user as parameter and returns whether they are busy currently or not. The `play` function takes the name of a user, plays the hold tone for one minute to them, and returns. The `connect` function takes two users as parameters, connects them if they are not busy, and returns when the call is disconnected. Identify the errors. Write the correct code.

```
hold_played = 0
while True:
  if hold_played >= 5:
    break
  if is_busy(callee):
    play_hold_tone(callee)
  connect(caller, callee)
  hold_played = hold_played + 1
```

2. (1 point) Consider the following fuction:

```
def foo(x, y):
  if x:
    if y:
      print("a")
    else:
      print("b")
  elif not y:
    print("c")
  elif not x:
    print("d")
  else:
    print("e")
```

   What should be the value of x and y to print the following? If it is not possible to print it, state and explain why.

   - a
   - b
   - c
   - d
   - e

3. (2 points) Write a function that takes a list of integers and a function on integers and returns the element in the list that maximizes the function. For example, if the list is `[-1, 1, -10, 5]` and the function is `abs`, then the function should return `-10`. For the empty list, the function should return `None`.

4. (1 point) Explain why the following two fragments of code are not equivalent (x and y are some Boolean values). Write a fragment of code equivalent to the first one without using nested if (or other nested blocks).

```
if x:
  if y:
    print("a")
  else:
    print("b")
else:
  print("c")
```

```
if x and y:
  print("a")
if not y:
  print("b")
else:
  print("c")
```

5. (2 points) For each of the following lines, write whether the condition is True, False, or an Error in Python.

```
10 > 12.1
"1234" > "234"
22 > "33"
"123456789"[3:7] < "522"
1 == 0 and 1/0 > 1
1 != 0 and 1/0 > 1
1 == 0 or 1/0 > 1
1 != 0 or 1/0 > 1
```

6. (2 points) List all possible outputs for this program along with the conditions on integers x and y that leads to that output. Each possible output should occur exactly once in your list. For example, ”a” is printed exactly when $x \in [11,\infty]$ and $y \in [-\infty,5]$.

```
if x > 10:
  if y < 6:
    print("a")
  elif x < 12:
    print("b")
  else:
    print("c")
elif y > 1:
  if x > 5:
    print("d")
  if y > 3:
    print("e")
else:
  print("e")
```

B 10 POINTS

Full Name ________________

Section & Subsection ________________

Roll # ________________

1. (2 points) The following fragment of code was intended to be part of a telephone service provider (like BSNL). Initially, the exchange checks if the receiver of the call is busy, and if so must play a hold tone for the caller and retry the same process after playing it. The hold tone should be played for at most 5 minutes. If receiver is not busy, the caller should be connected to the receiver and the program terminates after disconnection.

   The `is_busy` function takes a user as parameter and returns whether they are busy currently or not. The `play` function takes the name of a user, plays the hold tone for one minute to them, and returns. The `connect` function takes two users as parameters, connects them if they are not busy, and returns when the call is disconnected. Identify the errors. Write the correct code.

```
hold_played = 0
while True:
  if hold_played >= 5:
    break
  if is_busy(callee):
    play_hold_tone(callee)
  connect(caller, callee)
  hold_played = hold_played + 1
```

2. (2 points) For each of the following lines, write whether the condition is True, False, or an Error in Python.

```
10 > 12.1
"1234" > "234"
22 > "33"
"123456789"[3:7] < "522"
1 == 0 and 1/0 > 1
1 != 0 and 1/0 > 1
1 == 0 or 1/0 > 1
1 != 0 or 1/0 > 1
```

3. (2 points) List all possible outputs for this program along with the conditions on integers x and y that leads to that output. Each possible output should occur exactly once in your list. For example, ”a” is printed exactly when $x \in [11,\infty]$ and $y \in [-\infty,5]$.

```
if x > 10:
  if y < 6:
    print("a")
  elif x < 12:
    print("b")
  else:
    print("c")
elif y > 1:
  if x > 5:
    print("d")
  if y > 3:
    print("e")
else:
  print("e")
```

4. (1 point) Consider the following fuction:

```
def foo(x, y):
  if x:
    if y:
      print("a")
    else:
      print("b")
  elif not y:
    print("c")
  elif not x:
    print("d")
  else:
    print("e")
```

   What should be the value of x and y to print the following? If it is not possible to print it, state and explain why.

   - a
   - b
   - c
   - d
   - e

5. (2 points) Write a function that takes a list of integers and a function on integers and returns the element in the list that maximizes the function. For example, if the list is `[-1, 1, -10, 5]` and the function is `abs`, then the function should return `-10`. For the empty list, the function should return `None`.

6. (1 point) Explain why the following two fragments of code are not equivalent (x and y are some Boolean values). Write a fragment of code equivalent to the first one without using nested if (or other nested blocks).

```
if x:
  if y:                            if x and y:
    print("a")                       print("a")
  else:                            if not y:
    print("b")                       print("b")
else:                              else:
  print("c")                         print("c")
```

C 10 POINTS

Full Name ________________

Section & Subsection ________________

Roll # ________________

1. (1 point) Consider the following fuction:

```
def foo(x, y):
  if x:
    if y:
      print("a")
    else:
      print("b")
  elif not y:
    print("c")
  elif not x:
    print("d")
  else:
    print("e")
```

What should be the value of x and y to print the following? If it is not possible to print it, state and explain why.

- a
- b
- c
- d
- e

2. (2 points) Write a function that takes a list of integers and a function on integers and returns the element in the list that maximizes the function. For example, if the list is `[-1, 1, -10, 5]` and the function is `abs`, then the function should return `-10`. For the empty list, the function should return `None`.

3. (1 point) Explain why the following two fragments of code are not equivalent (x and y are some Boolean values). Write a fragment of code equivalent to the first one without using nested if (or other nested blocks).

```
if x:
  if y:
    print("a")
  else:
    print("b")
else:
  print("c")
```

```
if x and y:
  print("a")
if not y:
  print("b")
else:
  print("c")
```

4. (2 points) For each of the following lines, write whether the condition is True, False, or an Error in Python.

```
10 > 12.1
"1234" > "234"
22 > "33"
"123456789"[3:7] < "522"
1 == 0 and 1/0 > 1
1 != 0 and 1/0 > 1
1 == 0 or 1/0 > 1
1 != 0 or 1/0 > 1
```

5. (2 points) List all possible outputs for this program along with the conditions on integers x and y that leads to that output. Each possible output should occur exactly once in your list. For example, ”a” is printed exactly when $x \in [11,\infty]$ and $y \in [-\infty,5]$.

```
if x > 10:
  if y < 6:
    print("a")
  elif x < 12:
    print("b")
  else:
    print("c")
elif y > 1:
  if x > 5:
    print("d")
  if y > 3:
    print("e")
else:
  print("e")
```

6. (2 points) The following fragment of code was intended to be part of a telephone service provider (like BSNL). Initially, the exchange checks if the receiver of the call is busy, and if so must play a hold tone for the caller and retry the same process after playing it. The hold tone should be played for at most 5 minutes. If receiver is not busy, the caller should be connected to the receiver and the program terminates after disconnection.

The `is_busy` function takes a user as parameter and returns whether they are busy currently or not. The `play` function takes the name of a user, plays the hold tone for one minute to them, and returns. The `connect` function takes two users as parameters, connects them if they are not busy, and returns when the call is disconnected. Identify the errors. Write the correct code.

```
hold_played = 0
while True:
  if hold_played >= 5:
    break
  if is_busy(callee):
    play_hold_tone(callee)
  connect(caller, callee)
  hold_played = hold_played + 1
```

D 10 POINTS

Full Name ________________

Section & Subsection ________________

Roll # ________________

1. (2 points) List all possible outputs for this program along with the conditions on integers x and y that leads to that output. Each possible output should occur exactly once in your list. For example, ”a” is printed exactly when $x\in[11,\infty]$ and $y\in[-\infty,5]$.

```
if x > 10:
  if y < 6:
    print("a")
  elif x < 12:
    print("b")
  else:
    print("c")
elif y > 1:
  if x > 5:
    print("d")
  if y > 3:
    print("e")
else:
  print("e")
```

2. (1 point) Consider the following fuction:

```
def foo(x, y):
  if x:
    if y:
      print("a")
    else:
      print("b")
  elif not y:
    print("c")
  elif not x:
    print("d")
  else:
    print("e")
```

   What should be the value of x and y to print the following? If it is not possible to print it, state and explain why.

   - a
   - b
   - c
   - d
   - e

3. (2 points) The following fragment of code was intended to be part of a telephone service provider (like BSNL). Initially, the exchange checks if the receiver of the call is busy, and if so must play a hold tone for the caller and retry the same process after playing it. The hold tone should be played for at most 5 minutes. If receiver is not busy, the caller should be connected to the receiver and the program terminates after disconnection.

   The `is_busy` function takes a user as parameter and returns whether they are busy currently or not. The `play` function takes the name of a user, plays the hold tone for one minute to them, and returns. The `connect` function takes two users as parameters, connects them if they are not busy, and returns when the call is disconnected. Identify the errors. Write the correct code.

```
hold_played = 0
while True:
  if hold_played >= 5:
    break
  if is_busy(callee):
    play_hold_tone(callee)
  connect(caller, callee)
  hold_played = hold_played + 1
```

4. (2 points) Write a function that takes a list of integers and a function on integers and returns the element in the list that maximizes the function. For example, if the list is `[-1, 1, -10, 5]` and the function is **`abs`**, then the function should return `-10`. For the empty list, the function should return `None`.

5. (1 point) Explain why the following two fragments of code are not equivalent (x and y are some Boolean values). Write a fragment of code equivalent to the first one without using nested if (or other nested blocks).

```
if x:
  if y:                              if x and y:
    print("a")                         print("a")
  else:                              if not y:
    print("b")                         print("b")
else:                                else:
  print("c")                           print("c")
```

6. (2 points) For each of the following lines, write whether the condition is True, False, or an Error in Python.

```
10 > 12.1
"1234" > "234"
22 > "33"
"123456789"[3:7] < "522"
1 == 0 and 1/0 > 1
1 != 0 and 1/0 > 1
1 == 0 or 1/0 > 1
1 != 0 or 1/0 > 1
```

E 10 POINTS

Full Name ________________

Section & Subsection ________________

Roll # ________________

1. (2 points) For each of the following lines, write whether the condition is True, False, or an Error in Python.

```
10 > 12.1
"1234" > "234"
22 > "33"
"123456789"[3:7] < "522"
1 == 0 and 1/0 > 1
1 != 0 and 1/0 > 1
1 == 0 or 1/0 > 1
1 != 0 or 1/0 > 1
```

2. (1 point) Consider the following fuction:

```
def foo(x, y):
  if x:
    if y:
      print("a")
    else:
      print("b")
  elif not y:
    print("c")
  elif not x:
    print("d")
  else:
    print("e")
```

What should be the value of x and y to print the following? If it is not possible to print it, state and explain why.

- a
- b
- c
- d
- e

3. (2 points) Write a function that takes a list of integers and a function on integers and returns the element in the list that maximizes the function. For example, if the list is `[-1, 1, -10, 5]` and the function is `abs`, then the function should return `-10`. For the empty list, the function should return `None`.

4. (1 point) Explain why the following two fragments of code are not equivalent (x and y are some Boolean values). Write a fragment of code equivalent to the first one without using nested if (or other nested blocks).

```
if x:
  if y:                          if x and y:
    print("a")                     print("a")
  else:                          if not y:
    print("b")                     print("b")
else:                            else:
  print("c")                       print("c")
```

5. (2 points) List all possible outputs for this program along with the conditions on integers x and y that leads to that output. Each possible output should occur exactly once in your list. For example, ”a” is printed exactly when $x \in [11,\infty]$ and $y \in [-\infty,5]$.

```
if x > 10:
  if y < 6:
    print("a")
  elif x < 12:
    print("b")
  else:
    print("c")
elif y > 1:
  if x > 5:
    print("d")
  if y > 3:
    print("e")
else:
  print("e")
```

6. (2 points) The following fragment of code was intended to be part of a telephone service provider (like BSNL). Initially, the exchange checks if the receiver of the call is busy, and if so must play a hold tone for the caller and retry the same process after playing it. The hold tone should be played for at most 5 minutes. If receiver is not busy, the caller should be connected to the receiver and the program terminates after disconnection.

The `is_busy` function takes a user as parameter and returns whether they are busy currently or not. The `play` function takes the name of a user, plays the hold tone for one minute to them, and returns. The `connect` function takes two users as parameters, connects them if they are not busy, and returns when the call is disconnected. Identify the errors. Write the correct code.

```
hold_played = 0
while True:
  if hold_played >= 5:
    break
  if is_busy(callee):
    play_hold_tone(callee)
  connect(caller, callee)
  hold_played = hold_played + 1
```

F 10 POINTS

Full Name ________________

Section & Subsection ________________

Roll # ________________

1. (2 points) The following fragment of code was intended to be part of a telephone service provider (like BSNL). Initially, the exchange checks if the receiver of the call is busy, and if so must play a hold tone for the caller and retry the same process after playing it. The hold tone should be played for at most 5 minutes. If receiver is not busy, the caller should be connected to the receiver and the program terminates after disconnection.

   The `is_busy` function takes a user as parameter and returns whether they are busy currently or not. The `play` function takes the name of a user, plays the hold tone for one minute to them, and returns. The `connect` function takes two users as parameters, connects them if they are not busy, and returns when the call is disconnected. Identify the errors. Write the correct code.

```
hold_played = 0
while True:
  if hold_played >= 5:
    break
  if is_busy(callee):
    play_hold_tone(callee)
  connect(caller, callee)
  hold_played = hold_played + 1
```

2. (2 points) List all possible outputs for this program along with the conditions on integers x and y that leads to that output. Each possible output should occur exactly once in your list. For example, ”a” is printed exactly when $x \in [11,\infty]$ and $y \in [-\infty,5]$.

```
if x > 10:
  if y < 6:
    print("a")
  elif x < 12:
    print("b")
  else:
    print("c")
elif y > 1:
  if x > 5:
    print("d")
  if y > 3:
    print("e")
else:
  print("e")
```

3. (1 point) Consider the following fuction:

```
def foo(x, y):
  if x:
    if y:
      print("a")
    else:
      print("b")
  elif not y:
    print("c")
  elif not x:
    print("d")
  else:
    print("e")
```

   What should be the value of x and y to print the following? If it is not possible to print it, state and explain why.

   - a
   - b
   - c
   - d
   - e

4. (2 points) For each of the following lines, write whether the condition is True, False, or an Error in Python.

```
10 > 12.1
"1234" > "234"
22 > "33"
"123456789"[3:7] < "522"
1 == 0 and 1/0 > 1
1 != 0 and 1/0 > 1
1 == 0 or 1/0 > 1
1 != 0 or 1/0 > 1
```

5. (2 points) Write a function that takes a list of integers and a function on integers and returns the element in the list that maximizes the function. For example, if the list is `[-1, 1, -10, 5]` and the function is **`abs`**, then the function should return `-10`. For the empty list, the function should return `None`.

6. (1 point) Explain why the following two fragments of code are not equivalent (x and y are some Boolean values). Write a fragment of code equivalent to the first one without using nested if (or other nested blocks).

```
if x:
  if y:
    print("a")
  else:
    print("b")
else:
  print("c")
```

```
if x and y:
  print("a")
if not y:
  print("b")
else:
  print("c")
```

G 10 POINTS

Full Name ________________

Section & Subsection ________________

Roll # ________________

1. (1 point) Explain why the following two fragments of code are not equivalent (x and y are some Boolean values). Write a fragment of code equivalent to the first one without using nested if (or other nested blocks).

```
if x:
  if y:
    print("a")
  else:
    print("b")
else:
  print("c")
```

```
if x and y:
  print("a")
if not y:
  print("b")
else:
  print("c")
```

2. (2 points) List all possible outputs for this program along with the conditions on integers x and y that leads to that output. Each possible output should occur exactly once in your list. For example, ”a” is printed exactly when $x \in [11,\infty]$ and $y \in [-\infty,5]$.

```
if x > 10:
  if y < 6:
    print("a")
  elif x < 12:
    print("b")
  else:
    print("c")
elif y > 1:
  if x > 5:
    print("d")
  if y > 3:
    print("e")
else:
  print("e")
```

3. (2 points) For each of the following lines, write whether the condition is True, False, or an Error in Python.

```
10 > 12.1
"1234" > "234"
22 > "33"
"123456789"[3:7] < "522"
1 == 0 and 1/0 > 1
1 != 0 and 1/0 > 1
1 == 0 or 1/0 > 1
1 != 0 or 1/0 > 1
```

4. (2 points) Write a function that takes a list of integers and a function on integers and returns the element in the list that maximizes the function. For example, if the list is `[-1, 1, -10, 5]` and the function is `abs`, then the function should return `-10`. For the empty list, the function should return `None`.

5. (2 points) The following fragment of code was intended to be part of a telephone service provider (like BSNL). Initially, the exchange checks if the receiver of the call is busy, and if so must play a hold tone for the caller and retry the same process after playing it. The hold tone should be played for at most 5 minutes. If receiver is not busy, the caller should be connected to the receiver and the program terminates after disconnection.

   The `is_busy` function takes a user as parameter and returns whether they are busy currently or not. The `play` function takes the name of a user, plays the hold tone for one minute to them, and returns. The `connect` function takes two users as parameters, connects them if they are not busy, and returns when the call is disconnected. Identify the errors. Write the correct code.

```
hold_played = 0
while True:
  if hold_played >= 5:
    break
  if is_busy(callee):
    play_hold_tone(callee)
  connect(caller, callee)
  hold_played = hold_played + 1
```

6. (1 point) Consider the following fuction:

```
def foo(x, y):
  if x:
    if y:
      print("a")
    else:
      print("b")
  elif not y:
    print("c")
  elif not x:
    print("d")
  else:
    print("e")
```

   What should be the value of x and y to print the following? If it is not possible to print it, state and explain why.

   - a
   - b
   - c
   - d
   - e

H 10 POINTS

Full Name ________________

Section & Subsection ________________

Roll # ________________

1. (2 points) Write a function that takes a list of integers and a function on integers and returns the element in the list that maximizes the function. For example, if the list is `[-1, 1, -10, 5]` and the function is **`abs`**, then the function should return `-10`. For the empty list, the function should return `None`.

2. (2 points) For each of the following lines, write whether the condition is True, False, or an Error in Python.

```
10 > 12.1
"1234" > "234"
22 > "33"
"123456789"[3:7] < "522"
1 == 0 and 1/0 > 1
1 != 0 and 1/0 > 1
1 == 0 or 1/0 > 1
1 != 0 or 1/0 > 1
```

3. (2 points) List all possible outputs for this program along with the conditions on integers x and y that leads to that output. Each possible output should occur exactly once in your list. For example, ”a” is printed exactly when $x\in[11,\infty]$ and $y\in[-\infty,5]$.

```
if x > 10:
  if y < 6:
    print("a")
  elif x < 12:
    print("b")
  else:
    print("c")
elif y > 1:
  if x > 5:
    print("d")
  if y > 3:
    print("e")
else:
  print("e")
```

4. (1 point) Consider the following fuction:

```
def foo(x, y):
  if x:
    if y:
      print("a")
    else:
      print("b")
  elif not y:
    print("c")
  elif not x:
    print("d")
  else:
    print("e")
```

What should be the value of x and y to print the following? If it is not possible to print it, state and explain why.

- a
- b
- c
- d
- e

5. (1 point) Explain why the following two fragments of code are not equivalent (x and y are some Boolean values). Write a fragment of code equivalent to the first one without using nested if (or other nested blocks).

```
if x:
  if y:
    print("a")
  else:
    print("b")
else:
  print("c")
```

```
if x and y:
  print("a")
if not y:
  print("b")
else:
  print("c")
```

6. (2 points) The following fragment of code was intended to be part of a telephone service provider (like BSNL). Initially, the exchange checks if the receiver of the call is busy, and if so must play a hold tone for the caller and retry the same process after playing it. The hold tone should be played for at most 5 minutes. If receiver is not busy, the caller should be connected to the receiver and the program terminates after disconnection.

The `is_busy` function takes a user as parameter and returns whether they are busy currently or not. The `play` function takes the name of a user, plays the hold tone for one minute to them, and returns. The `connect` function takes two users as parameters, connects them if they are not busy, and returns when the call is disconnected. Identify the errors. Write the correct code.

```
hold_played = 0
while True:
  if hold_played >= 5:
    break
  if is_busy(callee):
    play_hold_tone(callee)
  connect(caller, callee)
  hold_played = hold_played + 1
```

I 10 POINTS

Full Name ________________

Section & Subsection ________________

Roll # ________________

1. (2 points) Write a function that takes a list of integers and a function on integers and returns the element in the list that maximizes the function. For example, if the list is `[-1, 1, -10, 5]` and the function is **`abs`**, then the function should return `-10`. For the empty list, the function should return `None`.

2. (1 point) Consider the following fuction:

```
def foo(x, y):
  if x:
    if y:
      print("a")
    else:
      print("b")
  elif not y:
    print("c")
  elif not x:
    print("d")
  else:
    print("e")
```

What should be the value of x and y to print the following? If it is not possible to print it, state and explain why.

- a
- b
- c
- d
- e

3. (2 points) The following fragment of code was intended to be part of a telephone service provider (like BSNL). Initially, the exchange checks if the receiver of the call is busy, and if so must play a hold tone for the caller and retry the same process after playing it. The hold tone should be played for at most 5 minutes. If receiver is not busy, the caller should be connected to the receiver and the program terminates after disconnection.

The `is_busy` function takes a user as parameter and returns whether they are busy currently or not. The `play` function takes the name of a user, plays the hold tone for one minute to them, and returns. The `connect` function takes two users as parameters, connects them if they are not busy, and returns when the call is disconnected. Identify the errors. Write the correct code.

```
hold_played = 0
while True:
  if hold_played >= 5:
    break
  if is_busy(callee):
    play_hold_tone(callee)
  connect(caller, callee)
  hold_played = hold_played + 1
```

4. (1 point) Explain why the following two fragments of code are not equivalent (x and y are some Boolean values). Write a fragment of code equivalent to the first one without using nested if (or other nested blocks).

```
if x:
  if y:                            if x and y:
    print("a")                       print("a")
  else:                            if not y:
    print("b")                       print("b")
else:                              else:
  print("c")                         print("c")
```

5. (2 points) For each of the following lines, write whether the condition is True, False, or an Error in Python.

```
10 > 12.1
"1234" > "234"
22 > "33"
"123456789"[3:7] < "522"
1 == 0 and 1/0 > 1
1 != 0 and 1/0 > 1
1 == 0 or 1/0 > 1
1 != 0 or 1/0 > 1
```

6. (2 points) List all possible outputs for this program along with the conditions on integers x and y that leads to that output. Each possible output should occur exactly once in your list. For example, ”a” is printed exactly when $x \in [11,\infty]$ and $y \in [-\infty,5]$.

```
if x > 10:
  if y < 6:
    print("a")
  elif x < 12:
    print("b")
  else:
    print("c")
elif y > 1:
  if x > 5:
    print("d")
  if y > 3:
    print("e")
else:
  print("e")
```

J 10 POINTS

Full Name ________________

Section & Subsection ________________

Roll # ________________

1. (2 points) For each of the following lines, write whether the condition is True, False, or an Error in Python.

```
10 > 12.1
"1234" > "234"
22 > "33"
"123456789"[3:7] < "522"
1 == 0 and 1/0 > 1
1 != 0 and 1/0 > 1
1 == 0 or 1/0 > 1
1 != 0 or 1/0 > 1
```

2. (1 point) Explain why the following two fragments of code are not equivalent (x and y are some Boolean values). Write a fragment of code equivalent to the first one without using nested if (or other nested blocks).

```
if x:
  if y:
    print("a")
  else:
    print("b")
else:
  print("c")
```

```
if x and y:
  print("a")
if not y:
  print("b")
else:
  print("c")
```

3. (2 points) List all possible outputs for this program along with the conditions on integers x and y that leads to that output. Each possible output should occur exactly once in your list. For example, ”a” is printed exactly when $x \in [11,\infty]$ and $y \in [-\infty,5]$.

```
if x > 10:
  if y < 6:
    print("a")
  elif x < 12:
    print("b")
  else:
    print("c")
elif y > 1:
  if x > 5:
    print("d")
  if y > 3:
    print("e")
else:
  print("e")
```

4. (1 point) Consider the following fuction:

```
def foo(x, y):
  if x:
    if y:
      print("a")
    else:
      print("b")
  elif not y:
    print("c")
  elif not x:
    print("d")
  else:
    print("e")
```

What should be the value of x and y to print the following? If it is not possible to print it, state and explain why.

- a
- b
- c
- d
- e

5. (2 points) The following fragment of code was intended to be part of a telephone service provider (like BSNL). Initially, the exchange checks if the receiver of the call is busy, and if so must play a hold tone for the caller and retry the same process after playing it. The hold tone should be played for at most 5 minutes. If receiver is not busy, the caller should be connected to the receiver and the program terminates after disconnection.

The `is_busy` function takes a user as parameter and returns whether they are busy currently or not. The `play` function takes the name of a user, plays the hold tone for one minute to them, and returns. The `connect` function takes two users as parameters, connects them if they are not busy, and returns when the call is disconnected. Identify the errors. Write the correct code.

```
hold_played = 0
while True:
  if hold_played >= 5:
    break
  if is_busy(callee):
    play_hold_tone(callee)
  connect(caller, callee)
  hold_played = hold_played + 1
```

6. (2 points) Write a function that takes a list of integers and a function on integers and returns the element in the list that maximizes the function. For example, if the list is `[-1, 1, -10, 5]` and the function is **`abs`**, then the function should return `-10`. For the empty list, the function should return `None`.

K 10 POINTS

Full Name ________________

Section & Subsection ________________

Roll # ________________

1. (2 points) Write a function that takes a list of integers and a function on integers and returns the element in the list that maximizes the function. For example, if the list is `[-1, 1, -10, 5]` and the function is **`abs`**, then the function should return `-10`. For the empty list, the function should return `None`.

2. (2 points) For each of the following lines, write whether the condition is True, False, or an Error in Python.

```
10 > 12.1
"1234" > "234"
22 > "33"
"123456789"[3:7] < "522"
1 == 0 and 1/0 > 1
1 != 0 and 1/0 > 1
1 == 0 or 1/0 > 1
1 != 0 or 1/0 > 1
```

3. (2 points) List all possible outputs for this program along with the conditions on integers x and y that leads to that output. Each possible output should occur exactly once in your list. For example, ”a” is printed exactly when $x \in [11,\infty]$ and $y \in [-\infty,5]$.

```
if x > 10:
  if y < 6:
    print("a")
  elif x < 12:
    print("b")
  else:
    print("c")
elif y > 1:
  if x > 5:
    print("d")
  if y > 3:
    print("e")
else:
  print("e")
```

4. (2 points) The following fragment of code was intended to be part of a telephone service provider (like BSNL). Initially, the exchange checks if the receiver of the call is busy, and if so must play a hold tone for the caller and retry the same process after playing it. The hold tone should be played for at most 5 minutes. If receiver is not busy, the caller should be connected to the receiver and the program terminates after disconnection.

   The `is_busy` function takes a user as parameter and returns whether they are busy currently or not. The `play` function takes the name of a user, plays the hold tone for one minute to them, and returns. The `connect` function takes two users as parameters, connects them if they are not busy, and returns when the call is disconnected. Identify the errors. Write the correct code.

```
hold_played = 0
while True:
  if hold_played >= 5:
    break
  if is_busy(callee):
    play_hold_tone(callee)
  connect(caller, callee)
  hold_played = hold_played + 1
```

5. (1 point) Explain why the following two fragments of code are not equivalent (x and y are some Boolean values). Write a fragment of code equivalent to the first one without using nested if (or other nested blocks).

```
if x:
  if y:                             if x and y:
    print("a")                        print("a")
  else:                             if not y:
    print("b")                        print("b")
else:                               else:
  print("c")                          print("c")
```

6. (1 point) Consider the following fuction:

```
def foo(x, y):
  if x:
    if y:
      print("a")
    else:
      print("b")
  elif not y:
    print("c")
  elif not x:
    print("d")
  else:
    print("e")
```

   What should be the value of x and y to print the following? If it is not possible to print it, state and explain why.

   - a
   - b
   - c
   - d
   - e

L 10 POINTS

Full Name ________________

Section & Subsection ________________

Roll # ________________

1. (1 point) Explain why the following two fragments of code are not equivalent (x and y are some Boolean values). Write a fragment of code equivalent to the first one without using nested if (or other nested blocks).

```
if x:
  if y:
    print("a")
  else:
    print("b")
else:
  print("c")
```

```
if x and y:
  print("a")
if not y:
  print("b")
else:
  print("c")
```

2. (2 points) The following fragment of code was intended to be part of a telephone service provider (like BSNL). Initially, the exchange checks if the receiver of the call is busy, and if so must play a hold tone for the caller and retry the same process after playing it. The hold tone should be played for at most 5 minutes. If receiver is not busy, the caller should be connected to the receiver and the program terminates after disconnection.

   The `is_busy` function takes a user as parameter and returns whether they are busy currently or not. The `play` function takes the name of a user, plays the hold tone for one minute to them, and returns. The `connect` function takes two users as parameters, connects them if they are not busy, and returns when the call is disconnected. Identify the errors. Write the correct code.

```
hold_played = 0
while True:
  if hold_played >= 5:
    break
  if is_busy(callee):
    play_hold_tone(callee)
  connect(caller, callee)
  hold_played = hold_played + 1
```

3. (2 points) List all possible outputs for this program along with the conditions on integers x and y that leads to that output. Each possible output should occur exactly once in your list. For example, ”a” is printed exactly when $x \in [11, \infty]$ and $y \in [-\infty, 5]$.

```
if x > 10:
  if y < 6:
    print("a")
  elif x < 12:
    print("b")
  else:
    print("c")
elif y > 1:
  if x > 5:
    print("d")
  if y > 3:
    print("e")
else:
  print("e")
```

4. (2 points) Write a function that takes a list of integers and a function on integers and returns the element in the list that maximizes the function. For example, if the list is `[-1, 1, -10, 5]` and the function is `abs`, then the function should return `-10`. For the empty list, the function should return `None`.

5. (2 points) For each of the following lines, write whether the condition is True, False, or an Error in Python.

```
10 > 12.1
"1234" > "234"
22 > "33"
"123456789"[3:7] < "522"
1 == 0 and 1/0 > 1
1 != 0 and 1/0 > 1
1 == 0 or 1/0 > 1
1 != 0 or 1/0 > 1
```

6. (1 point) Consider the following fuction:

```
def foo(x, y):
  if x:
    if y:
      print("a")
    else:
      print("b")
  elif not y:
    print("c")
  elif not x:
    print("d")
  else:
    print("e")
```

   What should be the value of x and y to print the following? If it is not possible to print it, state and explain why.

   - a
   - b
   - c
   - d
   - e

M 10 POINTS

Full Name ________________

Section & Subsection ________________

Roll # ________________

1. (1 point) Explain why the following two fragments of code are not equivalent (x and y are some Boolean values). Write a fragment of code equivalent to the first one without using nested if (or other nested blocks).

```
if x:
  if y:
    print("a")
  else:
    print("b")
else:
  print("c")
```

```
if x and y:
  print("a")
if not y:
  print("b")
else:
  print("c")
```

2. (2 points) The following fragment of code was intended to be part of a telephone service provider (like BSNL). Initially, the exchange checks if the receiver of the call is busy, and if so must play a hold tone for the caller and retry the same process after playing it. The hold tone should be played for at most 5 minutes. If receiver is not busy, the caller should be connected to the receiver and the program terminates after disconnection.

   The `is_busy` function takes a user as parameter and returns whether they are busy currently or not. The `play` function takes the name of a user, plays the hold tone for one minute to them, and returns. The `connect` function takes two users as parameters, connects them if they are not busy, and returns when the call is disconnected. Identify the errors. Write the correct code.

```
hold_played = 0
while True:
  if hold_played >= 5:
    break
  if is_busy(callee):
    play_hold_tone(callee)
  connect(caller, callee)
  hold_played = hold_played + 1
```

3. (2 points) Write a function that takes a list of integers and a function on integers and returns the element in the list that maximizes the function. For example, if the list is `[-1, 1, -10, 5]` and the function is `abs`, then the function should return `-10`. For the empty list, the function should return `None`.

4. (2 points) For each of the following lines, write whether the condition is True, False, or an Error in Python.

```
10 > 12.1
"1234" > "234"
22 > "33"
"123456789"[3:7] < "522"
1 == 0 and 1/0 > 1
1 != 0 and 1/0 > 1
1 == 0 or 1/0 > 1
1 != 0 or 1/0 > 1
```

5. (2 points) List all possible outputs for this program along with the conditions on integers x and y that leads to that output. Each possible output should occur exactly once in your list. For example, ”a” is printed exactly when $x\in[11,\infty]$ and $y\in[-\infty,5]$.

```
if x > 10:
  if y < 6:
    print("a")
  elif x < 12:
    print("b")
  else:
    print("c")
elif y > 1:
  if x > 5:
    print("d")
  if y > 3:
    print("e")
else:
  print("e")
```

6. (1 point) Consider the following fuction:

```
def foo(x, y):
  if x:
    if y:
      print("a")
    else:
      print("b")
  elif not y:
    print("c")
  elif not x:
    print("d")
  else:
    print("e")
```

   What should be the value of x and y to print the following? If it is not possible to print it, state and explain why.

   - a
   - b
   - c
   - d
   - e

N 10 POINTS

Full Name ________________

Section & Subsection ________________

Roll # ________________

1. (2 points) The following fragment of code was intended to be part of a telephone service provider (like BSNL). Initially, the exchange checks if the receiver of the call is busy, and if so must play a hold tone for the caller and retry the same process after playing it. The hold tone should be played for at most 5 minutes. If receiver is not busy, the caller should be connected to the receiver and the program terminates after disconnection.

   The `is_busy` function takes a user as parameter and returns whether they are busy currently or not. The `play` function takes the name of a user, plays the hold tone for one minute to them, and returns. The `connect` function takes two users as parameters, connects them if they are not busy, and returns when the call is disconnected. Identify the errors. Write the correct code.

```
hold_played = 0
while True:
  if hold_played >= 5:
    break
  if is_busy(callee):
    play_hold_tone(callee)
  connect(caller, callee)
  hold_played = hold_played + 1
```

2. (2 points) For each of the following lines, write whether the condition is True, False, or an Error in Python.

```
10 > 12.1
"1234" > "234"
22 > "33"
"123456789"[3:7] < "522"
1 == 0 and 1/0 > 1
1 != 0 and 1/0 > 1
1 == 0 or 1/0 > 1
1 != 0 or 1/0 > 1
```

3. (2 points) List all possible outputs for this program along with the conditions on integers x and y that leads to that output. Each possible output should occur exactly once in your list. For example, ”a” is printed exactly when $x \in [11,\infty]$ and $y \in [-\infty,5]$.

```
if x > 10:
  if y < 6:
    print("a")
  elif x < 12:
    print("b")
  else:
    print("c")
elif y > 1:
  if x > 5:
    print("d")
  if y > 3:
    print("e")
else:
  print("e")
```

4. (1 point) Explain why the following two fragments of code are not equivalent (x and y are some Boolean values). Write a fragment of code equivalent to the first one without using nested if (or other nested blocks).

```
if x:
  if y:
    print("a")
  else:
    print("b")
else:
  print("c")
```

```
if x and y:
  print("a")
if not y:
  print("b")
else:
  print("c")
```

5. (1 point) Consider the following fuction:

```
def foo(x, y):
  if x:
    if y:
      print("a")
    else:
      print("b")
  elif not y:
    print("c")
  elif not x:
    print("d")
  else:
    print("e")
```

   What should be the value of x and y to print the following? If it is not possible to print it, state and explain why.

   - a
   - b
   - c
   - d
   - e

6. (2 points) Write a function that takes a list of integers and a function on integers and returns the element in the list that maximizes the function. For example, if the list is `[-1, 1, -10, 5]` and the function is **`abs`**, then the function should return `-10`. For the empty list, the function should return `None`.