TYPE-B:FLOW OF CONTROL:CH-9

1)Rewrite the following code fragment that saves on the number of comparisons:

```
if (a == 0) :
  print ("Zero")
if (a == 1) :
  print ("One")
if (a == 2) :
  print ("Two")
if (a == 3) :
  print ("Three")
```

sol:

```
if a == 0:
    print("Zero")
elif a == 1:
    print("One")
elif a == 2:
    print("Two")
elif a == 3:
    print("Three")
```

2) Under what conditions will this code fragment print "water"?

```
if temp < 32 :
   print ("ice")
elif temp < 212:
   print ("water")
else :
   print ("steam")
```

sol:

The code will print "water" when:

32≤temp<212

3) What is the output produced by the following code?

```
x = 1
if x > 3 :
   if x > 4 :
        print ("A", end = ' ')
   else :
        print ("B", end = ' ')
elif x < 2:
   if (x != 0):
        print ("C", end = ' ')
print ("D")
```

sol:

```
C D
```

4) What is the error in following code? Correct the code:

```
weather = 'raining'
if weather = 'sunny' :
  print ("wear sunblock")
elif weather = "snow":
  print ("going skiing")
else :
  print (weather)
```

sol:

The = operator is used for assignment, not comparison.
To compare values, you must use ==.

5) What is the output of the following lines of code?

```
if int('zero') == 0 :
  print ("zero")
elif str(0) == 'zero' :
  print (0)
elif str(0) == '0' :
  print (str(0))
else:
  print ("none of the above")
```

sol:

```
ERROR!
Traceback (most recent call last):
  File "<main.py>", line 1, in <module>
ValueError: invalid literal for int() with base 10: 'zero'
```

6) Find the errors in the code given below and correct the code:

```
if n == 0
  print ("zero")
elif : n == 1
  print ("one")
elif
  n == 2:
  print ("two")
else n == 3:
  print ("three")
```

sol:

- Missing colon (:) after if n == 0.
- Wrong syntax in elif : n == 1 → colon is in the wrong place.

- Wrong syntax in elif n == 2: → correct placement needed.
- else n == 3: → else does not take a condition; use elif instead.

7) What is following code doing? What would it print for input as 3?

```
n = int(input( "Enter an integer:" ))
if n < 1 :
  print ("invalid value")
else :
  for i in range(1, n + 1):
    print (i * i)
```

sol:

```
Enter an integer:3
1
4
9
```

8) How are following two code fragments different from one another? Also, predict the output of the following code fragments :

(a)

```
n = int(input( "Enter an integer:" ))
if n > 0 :
    for a in range(1, n + n ) :
        print (a / (n/2))
    else :
        print ("Now quiting")
```

(b)

```
n = int(input("Enter an integer:"))
if n > 0 :
    for a in range(1, n + n) :
        print (a / (n/2))
else :
    print ("Now quiting")
```

sol:

Difference Between the Two Code Fragments
In (a), the else is connected to the for loop.
"Now quitting" prints after the loop finishes.
In (b), the else is connected to the if statement.
"Now quitting" prints only if n ≤ 0.
For n > 0:
(a) → Loop runs and then prints "Now quitting".

(b) → Only the loop runs; "Now quitting" is not printed.
The key difference is where the else is attached (for loop vs if statement).

9) (a) Rewrite the following code fragment using for loop:

```
i = 100
while (i > 0) :
   print (i)
   i -= 3

     sol:
        for i in range(100, 0, -3):
           print(i)
```

9)(b) Rewrite the following code fragment using for loop:

```
while num > 0 :
   print (num % 10)
   num = num/10

     sol:
        for i in range(len(str(num))):
           print(num % 10)
           num = num // 10
```

9)(c) Rewrite the following code fragment using for loop:

```
while num > 0 :
   count += 1
   sum += num
      num –= 2
      if count == 10 :
         print (sum/float(count))
         break

     sol:
        count = 0
        sum = 0
        for _ in range(10):   # loop runs 10 times
           sum += num
           count += 1
           num -= 2
        print(sum / float(count))
```

10)(a) Rewrite following code fragment using while loops :

```
    min = 0
    max = num
    if num < 0 :
       min = num
       max = 0 # compute sum of integers
             # from min to max

       for i in range(min, max + 1):
          sum += i
```

```
sol:
  min_val = 0
  max_val = num
  sum = 0

  if num < 0:
    min_val = num
    max_val = 0

    i = min_val
    while i <= max_val:
      sum += i
      i += 1
```

10)(b) Rewrite following code fragment using while loops :

```
for i in range(1, 16) :
  if i % 3 == 0 :
    print (i)
```

```
sol:
  i = 1
  while i < 16:
    if i % 3 == 0:
      print(i)
    i += 1
```

10)(c) Rewrite following code fragment using while loops :

```
for i in range(4) :
  for j in range(5):
    if i + 1 == j or j + 1 == 4 :
      print ("+", end = ' ')
  else :
    print ("o", end = ' ')
print()
```

```
sol:
  i = 0
  while i < 4:
    j = 0
    while j < 5:
      if i + 1 == j or j + 1 == 4:
        print("+", end=' ')
      else:
        print("o", end=' ')
      j += 1
    print()
    i += 1
```

11)(a) Predict the output of the following code fragments:

```
count = 0
while count < 10:
   print ("Hello")
   count += 1
```

sol:

```
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
Hello
```

11)(b) Predict the output of the following code fragments:

```
x = 10
y = 0
while x > y:
  print (x, y)
  x = x - 1
  y = y + 1
```

sol:

```
10 0
9 1
8 2
7 3
6 4
```

11)(c) Predict the output of the following code fragments:

```
keepgoing = True
x=100
while keepgoing :
   print (x)
   x = x - 10
   if x < 50 :
      keepgoing = False
```

sol:

```
100
90
80
70
60
50
```

11)(d) Predict the output of the following code fragments:

```
x = 45
while x < 50 :
   print (x)
```

sol:

```
endless loop prints 45 through infinite
```

11)(e) Predict the output of the following code fragments:

```
for x in [1,2,3,4,5]:
   print (x)
```

sol:

```
1
2
3
4
5
```

11)(f) Predict the output of the following code fragments:

```
for x in range(5):
  print (x)
```

sol:

```
0
1
2
3
4
```

11)(g) Predict the output of the following code fragments:

```
for p in range(1,10):
  print (p)
```

sol:

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

11)(h) Predict the output of the following code fragments:

```
for q in range(100, 50, -10):
   print (q)

   sol:
      100
      90
      80
      70
      60
```

11)(i) Predict the output of the following code fragments:

```
for z in range(-500, 500, 100):
   print (z)

   sol:
      -500
      -400
      -300
      -200
      -100
      0
      100
      200
      300
      400
```

11)(j) Predict the output of the following code fragments:

```
for y in range(500, 100, 100):
   print (" * ", y)

   sol:
      no output
```

11)(k) Predict the output of the following code fragments:

```
x = 10
y = 5
for i in range(x-y * 2):
   print (" % ", i)

   sol:
      no output
```

11)(l) Predict the output of the following code fragments:

```
for x in [1,2,3]:
   for y in [4, 5, 6]:
      print (x, y)
```

sol:
```
1 4
1 5
1 6
2 4
2 5
2 6
3 4
3 5
3 6
```

11)(m) Predict the output of the following code fragments:

```
for x in range(3):
    for y in range(4):
        print (x, y, x + y)
```

sol:
```
0 0 0
0 1 1
0 2 2
0 3 3
1 0 1
1 1 2
1 2 3
1 3 4
2 0 2
2 1 3
2 2 4
2 3 5
```

11)(n) Predict the output of the following code fragments:

```
c = 0
for x in range(10):
    for y in range(5):
        c += 1
print (c)
```

sol:
```
50
```

12) What is the output of the following code?

```
for i in range(4):
  for j in range(5):
      if i + 1 == j or j + i == 4:
          print ("+", end = ' ')
      else:
          print ("o", end = ' ')
print()
```

sol:

```
o + o o + o o + + o o o + + o o + o o +
```

13) In the nested for loop code below, how many times is the condition of the if clause evaluated?

```
for i in range(4):
  for j in range(5):
      if i + 1 == j or j + i == 4:
         print ("+", end = '')
      else:
         print ("o", end = '')
print()
```

sol:

Outer loop: for i in range(4) → i = 0, 1, 2, 3 → 4 iterations

Inner loop: for j in range(5) → j = 0, 1, 2, 3, 4 → 5 iterations per i

14) ans: (b)

15) Find the error. Consider the following program :

```
a = int(input("Enter a value: "))
while a != 0:
   count = count + 1
   a = int(input("Enter a value: "))
print("You entered", count, "values.")
```

It is supposed to count the number of values entered by the user until the user enters 0 and then display the count (not including the 0). However, when the program is run, it crashes with the following error message after the first input value is read :

```
Enter a value: 14
Traceback (most recent call last):
  File "count.py", line 4, in <module>
    count = count + 1
NameError: name 'count' is not defined
```

What change should be made to this program so that it will run correctly ? Write the modification that is needed into the program above, crossing out any code that should be removed and clearly indicating where any new code should be inserted.

sol:

MODIFIED CODE

```
a = int(input("Enter a value: "))
count = 0      # <-- NEW CODE: initialize count
while a != 0:
   count = count + 1
   a = int(input("Enter a value: "))
print("You entered", count, "values.")
```