Given these assignments: `a = 5`, `b = 2`, and `s = 1.5` write the type and value of the following expressions. Circle `ERROR` if the expression will result in a run time error.

<table>
<thead>
<tr>
<th>Statement</th>
<th>Type</th>
<th>Value</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>a / b</code></td>
<td><code>float</code></td>
<td><code>2.5</code></td>
<td>ERROR</td>
</tr>
<tr>
<td><code>b ** a</code></td>
<td><code>int</code></td>
<td><code>32</code></td>
<td>ERROR</td>
</tr>
<tr>
<td><code>float(a) / b</code></td>
<td><code>float</code></td>
<td><code>2.5</code></td>
<td>ERROR</td>
</tr>
<tr>
<td><code>a % b</code></td>
<td><code>int</code></td>
<td><code>1</code></td>
<td>ERROR</td>
</tr>
<tr>
<td><code>s // a</code></td>
<td><code>float</code></td>
<td><code>0.0</code></td>
<td>ERROR</td>
</tr>
<tr>
<td><code>a =&gt; b</code></td>
<td>Error</td>
<td>Error</td>
<td>ERROR</td>
</tr>
<tr>
<td><code>a == b</code></td>
<td><code>bool</code></td>
<td><code>False</code></td>
<td>ERROR</td>
</tr>
<tr>
<td><code>a // b</code></td>
<td><code>int</code></td>
<td><code>2</code></td>
<td>ERROR</td>
</tr>
<tr>
<td><code>a + b * a</code></td>
<td><code>int</code></td>
<td><code>15</code></td>
<td>ERROR</td>
</tr>
</tbody>
</table>
2  Conversion between binary, decimal and hexadecimal numbers:

   a. Convert the binary number 11101011 to decimal: ________235_________

   b. Convert the decimal number 151 to binary: ___10010111___

   c. Convert the binary number 10011100 to hexadecimal: _______9C________

   d. Convert the hexadecimal number 5F to binary: _______1011111________ (please show all 8 binary digits)

   e. Convert the decimal number 90 to hexadecimal: _______5A__________

3  What is the output from the following code if the user enters 75?

   c = int(input('Enter a value: '))
   if c > 100:
       print("A")
   elif c > 50:
       if c % 5 == 0 and not(c % 10 == 0):
           print("B")
       elif c % 5 == 0:
           print("C")
       else:
           print("D")
   if c > 20:
       print("E")
   else:
       print("F")

   Your answer:  BE
4  What is the output from the following code?

```python
acc = 0
for i in range(5,15,5):
    var = i
    while var > 0:
        var //= 2
        acc += var
    print("i=",i," var=",var)
print("acc=",acc)
```

Your answer:

```
i=5 var=2
i=5 var=1
i=5 var=0
i=10 var=5
i=10 var=2
i=10 var=1
i=10 var=0
acc=11
```

5  Write a program that prompts for radius length. Your program should calculate and print the resulting circumference and area of a circle with that radius. You must also check that the given radius is positive. Otherwise, print an error message and do not carry out the calculations. Use pi = 3.14 for this question rather than importing math module.

Code:
def main():
    radius = float(input('Enter a radius: '))
    pi = 3.14
    if radius > 0:
        circumference = 2 * pi * radius
        area = pi * (radius ** 2)
        print(("Circumference: " + str(circle) + ", Area: " + str(area))
    else:
        print('ERROR: Radius must be positive')

6  Write a program that prompts a row number and print out the pattern in a zig-zag way. If the leading number of the row is odd, the row displays numbers in a decreasing sequence, starting from the leading number to 1. If the leading number of the row is even, the row displays numbers in an increasing sequence, starting from 1 to the leading number. Assume the input is always a valid positive integer.

Code:

Solution #1: Using for loop

row = int(input("Enter # of row: "))

for i in range (row, 0, -1):
    if i % 2 == 0:
        line = ""
        for j in range(1, i+1):
            line += str(j) + ""
        print(line)
    else:
        line = ""
for j in range(i, 0, -1):
    line += str(j) + ' ' 
print(line)

print()

Solution #2: Using while loop

row = int(input("Enter # of row: "))
while row != 0:
    if row % 2 == 0:
        temp_row = 1
        while temp_row <= row:
            print(temp_row, end = " ")
            temp_row += 1
    else:
        temp_row = row
        while temp_row != 0:
            print(temp_row, end = " ")
            temp_row -= 1
    print()
    row -= 1

7 Given a positive integer number, write a program to print the total number of times each digit Write a program that prompts the user to enter a sequence of positive integers where each integer represents how many hours the employee worked in a day this week. When the user enters a negative integer, there are no more days to input. However, since there is a maximum of 7 days in a week, so at most you can take inputs for 7 times. The program should then print out (a) the employee's bonus pay for that week; (b) the employee's overtime pay for that week; and (c) the employee's total pay for that week. (In the following examples, pay_rate is 10.0.)
The rules governing an employee's pay are as follows:

a. Each employee has an hourly pay rate, which we will call pay_rate. An employee is paid pay_rate dollars for every hour worked. pay_rate is a variable defined for you in advance; you should directly use it instead of defining it or read it in.

b. If an employee works more than 10 hours in a single day, they must be paid an additional bonus of $13 for each such day.

c. If an employee works a total of more than 40 hours in a single week, any hours over 40 will be paid at an overtime rate of one-and-a-half times their usual hourly wage. Hours under 40 will be paid at the usual rate. For example, if an employee has a normal rate of $10 per hour and works 45 hours in a single week, they will be paid $10 x 40 = $400 for the first 40 hours, then an additional overtime of 1.5 x $10 x 5 = $75 for the remaining 5 hours, for a total pay of $475.

The formatting and number of decimal places output in your calculations are not taken into account in grading your work.

**Code:**

```python
WEEKDAY_NUM = 7
REG_W_HOUR = 40
REG_D_HOUR = 10
BONUS_PAY = 13.0
OVERTIME_RATE = 1.5

day_hour = int(input("Enter the hours worked:
"))
total_hours = 0
bonus_num = 0
day_count = 1

while day_hour >= 0 and day_count <= WEEKDAY_NUM:
    if day_hour > REG_D_HOUR:
        bonus_num += 1
    total_hours += day_hour
```

CS 1114 Exam 1 Review
day_count += 1

if day_count <= WEEKDAY_NUM:
    day_hour = int(input())

if total_hours > REG_W_HOUR:
    overtime_pay = (total_hours - REG_W_HOUR) * OVERTIME_RATE * pay_rate
    bonus_pay = bonus_num * BONUS_PAY
    total = REG_W_HOUR * payRATE + overtime_pay + bonus_pay
else:
    overtime_pay = 0.0
    bonus_pay = bonus_num * BONUS_PAY
    total = total_hours * payRATE + bonus_pay

print("Bonus Pay: ", round(bonus_pay,2))
print("Overtime Pay: ", round(overtime_pay,2))
print("Total Pay: ", round(total,2))