Each question is worth 2 points. Feel free to use your computer to verify your answers.

For the following questions, use this string and code. Indicate if there are any leading or trailing spaces.

NOTE: Answers from 1 to 4 are enclosed in double quotes. Please consider the contents in the double quotes as the final answer.

```python
import re
str = '### Subject: scuba diving at... From: steve.millman@asu.
    edu Body: Underwater, where it is at'
match = re.search(r'XXX',str)
if match:
 print(match.group())
else:
 print('did not find')
```

1. **Replacing 'XXX' with ':.*:' , what prints out?**

   ": scuba diving at... From: steve.millman@asu.edu Body:"

2. **Replacing 'XXX' with 'st\S*' , what prints out?**

   "steve.millman@asu.edu"

3. **Replacing 'XXX' with ':.*?:' , what prints out?**

   ": scuba diving at... From:"

4. **Replacing 'XXX' with '[\w]*at$', what prints out?**

   " where it is at"

For the following questions, there is a string called my_text. You do not know the contents of my_text or how many words, sentences, or paragraphs it contains. You will provide the code to replace XXX to find the first instance of the requested text within my_text.

```python
import re
match = re.search(r'XXX',my_text)
```

How would you replace XXX to find:

1. **The shortest text between '$<$' and '$>$' including the angle brackets?**

   ANSWER: $< .*? >$

   ```python
   str = "#include <stdio.h> #include <bits/stdc++.h> int main()
      { int val = '<dfgt>'}"
   match = re.search('<.*?>', str)
   if match:
      print(match.group())
   else:
      print('did not find')
   ```

2. **The longest text between '<' and '>' including the angle brackets?**

   ANSWER: $< .* >$

   ```python
   str = "#include <stdio.h> #include <bits/stdc++.h> int main()
      { int val = '<dfgt>'}"
   match = re.search('<.*>', str)
   if match:
      print(match.group())
   else:
      print('did not find')
   ```

3. **A field of numbers if it is between MOD and X. For example, if MOD2357X is in the string, we want 2357. This should work for any number, not just the example number.**

   ANSWER: $(? <= MOD)\d + (? = X)$

   ```python
   str = "MOD2357X"
   match = re.search(r'(?<=MOD)\d+(?=X)', str)
   if match:
      print(match.group())
   else:
      print('did not find')
   ```

```
In [20]: str = "#include <stdio.h> #include <bits/stdc++.h> int main() { int val = '<dfgt>'}"
         match = re.search('<.*?>', str)
         if match:
             print(match.group())
         else:
             print('did not find')

         <stdio.h>
```

Figure 1: Question 1 of second set

```
In [21]: str = "#include <stdio.h> #include <bits/stdc++.h> int main() { int val = '<dfgt>'}"
         match = re.search('<.*>', str)
         if match:
             print(match.group())
         else:
             print('did not find')

         <stdio.h> #include <bits/stdc++.h> int main() { int val = '<dfgt>
```

Figure 2: Question 2 of second set

```
In [22]: str = "MOD2357X"

         # match = re.search(r'MOD([0-9]+)X', str)
         match = re.search(r'(?<=MOD)\d+(?=X)', str)
         if match:
             print(match.group())
         else:
             print('did not find')

         2357
```

Figure 3: Question 3 of second set