```
pip install PrettyTable
```

```
Collecting PrettyTable
  Downloading prettytable-3.5.0-py3-none-any.whl (26 kB)
Requirement already satisfied: wcwidth in c:\users\redol\anaconda3\
lib\site-packages (from PrettyTable) (0.2.5)
Installing collected packages: PrettyTable
Successfully installed PrettyTable-3.5.0
Note: you may need to restart the kernel to use updated packages.
```

```
from prettytable import PrettyTable
```

```
def while_loop(cleaned_code):
    final_code = []
    while_idx = None
    for i in range(len(cleaned_code)):
        codeline = cleaned_code[i]

        if 'while' in codeline:
            while_idx = i
            # The loop condition would be enclosed in brackets
            start_idx = codeline.index('(')
            end_idx = codeline.index(')')
            # Select the substring between start_idx and end_idx
            bool_condn = ''.join(codeline[start_idx:end_idx+1])
             # Replace with
            final_code.append('if !{}
goto({})'.format(bool_condn,None))
            while_idx = i
        elif '}' in codeline:
            final_code.append('goto({})'.format(while_idx+1))
            #
            final_code[while_idx] =
final_code[while_idx].replace('None',str(i+2))
            while_idx = None
        else:
            final_code.append(codeline)
    return final_code
```

```
with open('code.txt') as f:
    code = f.readlines()
    print('The Statement is:')
print(''.join(code))
cleaned_code = []
for i in range(len(code)):
    if code[i] != '\n':
        if code[i][-1] == '\n':
            # don't include the \n at the end of each line
            cleaned_code.append(code[i][:-1].strip())
        else:
            # strip() removes the trailing whitespaces on both ends of
```

```
string
            cleaned_code.append(code[i].strip())
```

```
The Statement is:
for(i=0, i>100, i++){
     c= c*200;
     d = c + 32;
}
```

```
intermediate_code = []
for i in range(len(cleaned_code)):
    codeline = cleaned_code[i]
    if 'for' in codeline:
        # for(init; condition; update1, update2, update3, etc.)\n
        conditions = codeline[4:-2].split(';')
        initialization = conditions[0].strip()
        break_condn = conditions[1].strip()
        updations = conditions[2].strip().split(',')
        intermediate_code.append(initialization)
        intermediate_code.append('while(' + break_condn + '){')
    elif '}' in codeline:
        for updation in updations:
            intermediate_code.append(updation+';')
        intermediate_code.append('}')
    else:
        intermediate_code.append(codeline)
```

```
----------------------------------------------------------------------
-----
IndexError                                Traceback (most recent call
last)
Input In [10], in <cell line: 2>()
      6 conditions = codeline[4:-2].split(';')
      7 initialization = conditions[0].strip()
----> 8 break_condn = conditions[1].strip()
      9 updations = conditions[2].strip().split(',')
     10 intermediate_code.append(initialization)

IndexError: list index out of range
```

```
final_code = while_loop(intermediate_code)

print('\nThe Three Code generated is:')
x1 = PrettyTable()
x1.field_names = ['Index','Code']
for i in range(len(final_code)):
    x1.add_row([i+1,final_code[i]])

print(x1)
```

```
The Three Code generated is:
+-------+------+
| Index | Code |
+-------+------+
+-------+------+
```