I hereby request a re-mark for my CSC190 midterm 2.

Specifically, I hope that my response to question 2, questions 4 part 4, question 5 part 4, and question 7 part 3. I will now provide my argument, with evidence when appropriate, that my response should be reconsidered.

Question 2:
My code for question 2 manipulates the bits correctly in each case. The only mistake I made was that I forgot an “else” before the second “if” statement. I have copied and compiled my original code and saw that my code gives the same output as the solution provided when n <=0. I then added the “else” and my code gives identical output as the solution. However, I only received credits for my “if” statements and my “for” loops, none for my bit manipulation. I have attached a print out of my code and the outputs and marked on it where the “else” was added. Thus, I believe that I should have only been penalized for one, or potentially the subsequent one, wrong conditional statement(s).

Question 4 part 4:
For this question, I thought that the statement was somewhat ambiguous. So I stated that I will consider two cases: one in which we have unsigned int x with 0 everywhere except in positions 5, 10, 15, and 31, the other case in which we are preserving the other bit information originally in x and only forcing some “1”’s in these positions. I provided code for both cases in my response. As it turns out, the first assumption fits well with the intention of the question, and my code provided with that assumption matches the solution. However, I did not receive full credit for this question. We are encouraged to state our assumptions when we are not sure what the question is asking, and I find it hard to take that I was penalized for covering my bases while providing the correct answer to the correct assumption.

Question 5 part 4:
When I wrote my adjacency list, I tried to represent every edge tuple/list use a pair of squares. Unfortunately, I was in a rush and my last edge for the C node looked like “M1”. I understand that it is a fault of my negligence as I did not clearly write “N | 1”. However, considering that every other edge entry was correct and there was an evident pattern in how I presented the edges, I hope I can receive more credit for this question.

Question 7 part 3:
In the adjacency matrix, I used 0 to present that distance between the same node (ie. Ag-Ag), and inf to represent the distance between non-adjacent nodes. The professor later confirmed in lecture that this should have been accepted as a correct response. I believe I lost marks because of the 0 entries along the main diagonal and I hope to receive full credit for this question.

I hope this request, and my arguments, find you well, and thank you for your time and consideration.

Juntai Luo

Edward L

Student number: 1003830526
email: edward.luo@mail.utoronto.ca

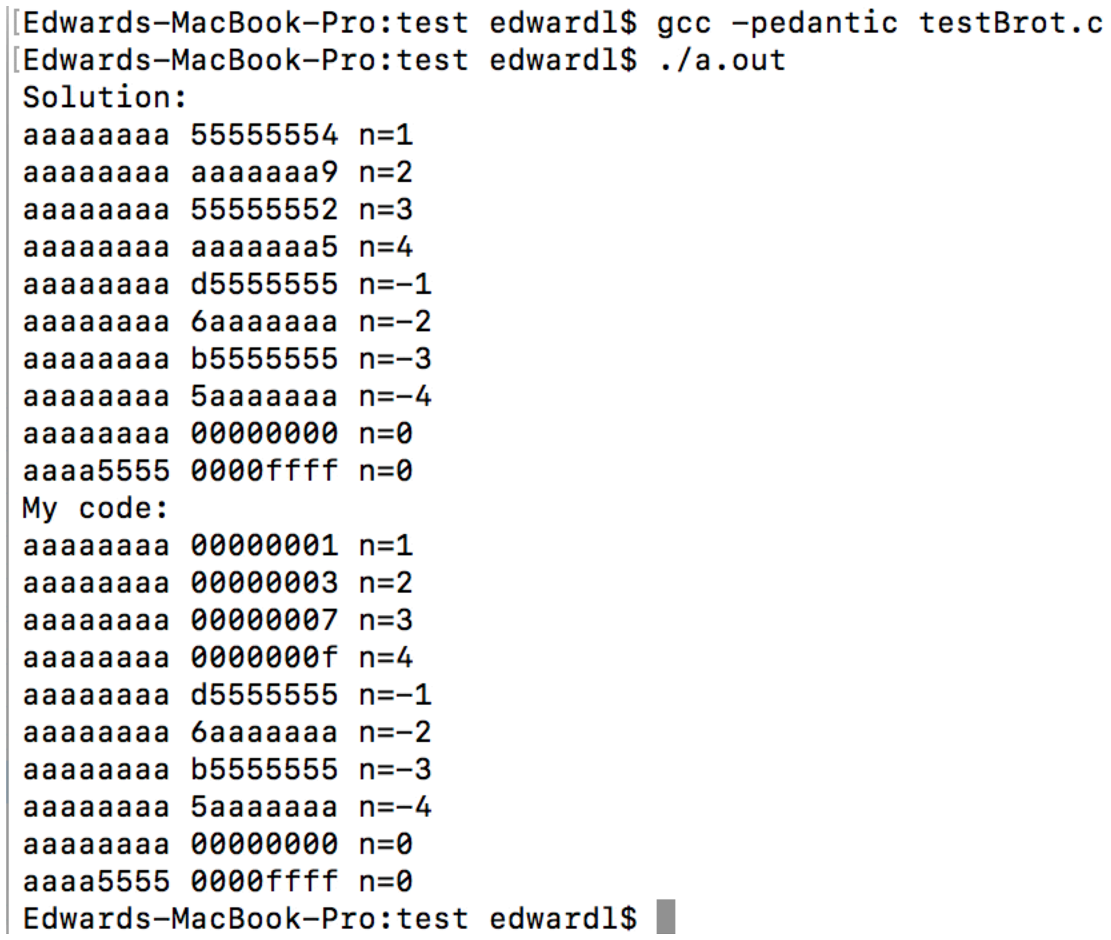
```c
#include <stdio.h>

unsigned int rot_soln(unsigned int x, int n){
  unsigned int accum = x;
  unsigned int shift_out;
  int i = 0;
  if (n>0){
    for (i = 0; i < n; i++) {
      shift_out = (~accum >> 31) & 0x1;
      accum = accum << 1;
      accum = accum | shift_out;
    }
  }else if (n<0) {
    for (i = 0; i < -n; i++) {
      shift_out = ~accum & 0x1;
      accum = accum >> 1;
      accum = accum | (shift_out << 31);}
  }else{
    accum = ((x>>16)&0xffff)^(x&0xffff);
  }
  return accum;
}

unsigned int rot(unsigned int x, int n){
  if (n>0) {
    unsigned int a, b;
    int c;
    a = 0x1;
    for (c=0;c<n;c++){
      a = a<<1;
      b= 0x1;
      a = a|b;}
    a = a<<(32-n);
    b = x & a;
    b = b>>(32-n);
    a = a>>(32-n);
    b = b^a;
    x = x<<n;
    x = x|b;
  } if (n<0) {
    unsigned int a, b;
    int c;
    a = 0x1;
    for(c=0;c<-n;c++){
      a = a<<1;
      b = 0x1;
      a = a|b;}
    b = x&a;
    b = b^a;
    b = b<<(32+n);
    x = x >> (-n);
    x = x|b;
  }else{
    unsigned int btm, top;
    btm = 0xFFFF;
    top = 0xFFFF0000;
    btm = x & btm;
    top = (x&top)>>16;
    x = btm^top;
  }
  return x;
}

int main(void) {
  unsigned int i;
  printf("Solution:\n");
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,1),1 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,2),2 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,3),3 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,4),4 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,-1),-1 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,-2),-2 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,-3),-3 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,-4),-4 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,0),0 );
  i = 0xaaaa5555; printf("%08x %08x n=%d\n",i,rot_soln(i,0),0 );
  /**/
  printf("My code:\n");
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,1),1 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,2),2 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,3),3 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,4),4 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,-1),-1 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,-2),-2 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,-3),-3 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,-4),-4 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,0),0 );
  i = 0xaaaa5555; printf("%08x %08x n=%d\n",i,rot(i,0),0 );
  return 0;
}
```

This is the output of the code



```
Edwards-MacBook-Pro:test edwardl$ gcc -pedantic testBrot.c
Edwards-MacBook-Pro:test edwardl$ ./a.out
Solution:
aaaaaaaa 55555554 n=1
aaaaaaaa aaaaaaa9 n=2
aaaaaaaa 55555552 n=3
aaaaaaaa aaaaaaa5 n=4
aaaaaaaa d5555555 n=-1
aaaaaaaa 6aaaaaaa n=-2
aaaaaaaa b5555555 n=-3
aaaaaaaa 5aaaaaaa n=-4
aaaaaaaa 00000000 n=0
aaaa5555 0000ffff n=0
My code:
aaaaaaaa 00000001 n=1
aaaaaaaa 00000003 n=2
aaaaaaaa 00000007 n=3
aaaaaaaa 0000000f n=4
aaaaaaaa d5555555 n=-1
aaaaaaaa 6aaaaaaa n=-2
aaaaaaaa b5555555 n=-3
aaaaaaaa 5aaaaaaa n=-4
aaaaaaaa 00000000 n=0
aaaa5555 0000ffff n=0
Edwards-MacBook-Pro:test edwardl$
```

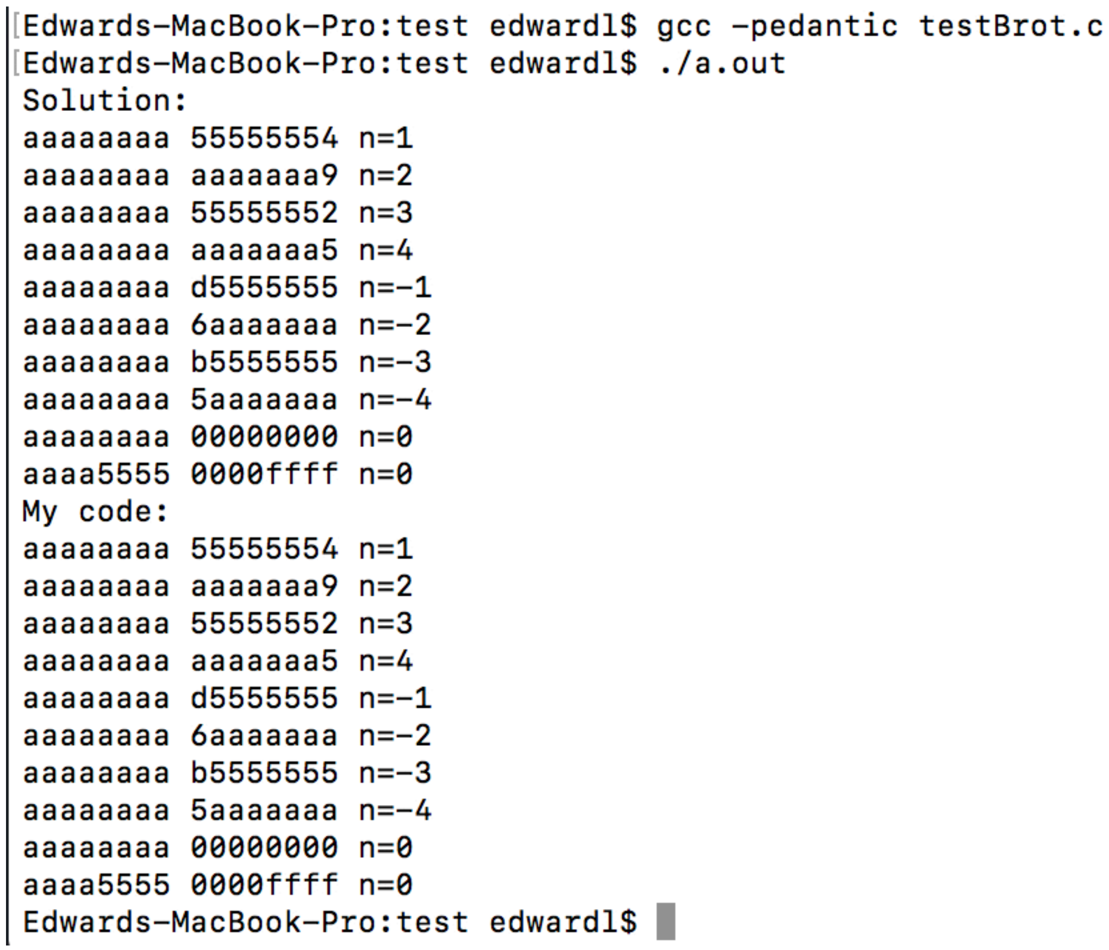
```
#include <stdio.h>

unsigned int rot_soln(unsigned int x, int n){
  unsigned int accum = x;
  unsigned int shift_out;
  int i = 0;
  if (n>0){
    for (i = 0; i < n; i++) {
      shift_out = (~accum >> 31) & 0x1;
      accum = accum << 1;
      accum = accum | shift_out;
    }
  }else if (n<0) {
    for (i = 0; i < -n; i++) {
      shift_out = ~accum & 0x1;
      accum = accum >> 1;
      accum = accum | (shift_out << 31);}
  }else{
    accum = ((x>>16)&0xffff)^(x&0xffff);
  }
  return accum;
}

unsigned int rot(unsigned int x, int n){
  if (n>0) {
    unsigned int a, b;
    int c;
    a = 0x1;
    for (c=0;c<n;c++){
      a = a<<1;
      b= 0x1;
      a = a|b;}
    a = a<<(32-n);
    b = x & a;
    b = b>>(32-n);
    a = a>>(32-n);
    b = b^a;
    x = x<<n;
    x = x|b;
  } else if (n<0) {                                   else is added here, only change made
    unsigned int a, b;
    int c;
    a = 0x1;
    for(c=0;c<-n;c++){
      a = a<<1;
      b = 0x1;
      a = a|b;}
    b = x&a;
    b = b^a;
    b = b<<(32+n);
    x = x >> (-n);
    x = x|b;
  }else{
    unsigned int btm, top;
    btm = 0xFFFF;
    top = 0xFFFF0000;
    btm = x & btm;
    top = (x&top)>>16;
    x = btm^top;
  }
  return x;
}

int main(void) {
  unsigned int i;
  printf("Solution:\n");
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,1),1 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,2),2 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,3),3 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,4),4 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,-1),-1 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,-2),-2 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,-3),-3 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,-4),-4 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot_soln(i,0),0 );
  i = 0xaaaa5555; printf("%08x %08x n=%d\n",i,rot_soln(i,0),0 );
  /**/
  printf("My code:\n");
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,1),1 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,2),2 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,3),3 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,4),4 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,-1),-1 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,-2),-2 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,-3),-3 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,-4),-4 );
  i = 0xaaaaaaaa; printf("%08x %08x n=%d\n",i,rot(i,0),0 );
  i = 0xaaaa5555; printf("%08x %08x n=%d\n",i,rot(i,0),0 );
  return 0;
}
```

This is the output of the code



```
[Edwards-MacBook-Pro:test edwardl$ gcc -pedantic testBrot.c
[Edwards-MacBook-Pro:test edwardl$ ./a.out
Solution:
aaaaaaaa 55555554 n=1
aaaaaaaa aaaaaaa9 n=2
aaaaaaaa 55555552 n=3
aaaaaaaa aaaaaaa5 n=4
aaaaaaaa d5555555 n=-1
aaaaaaaa 6aaaaaaa n=-2
aaaaaaaa b5555555 n=-3
aaaaaaaa 5aaaaaaa n=-4
aaaaaaaa 00000000 n=0
aaaa5555 0000ffff n=0
My code:
aaaaaaaa 55555554 n=1
aaaaaaaa aaaaaaa9 n=2
aaaaaaaa 55555552 n=3
aaaaaaaa aaaaaaa5 n=4
aaaaaaaa d5555555 n=-1
aaaaaaaa 6aaaaaaa n=-2
aaaaaaaa b5555555 n=-3
aaaaaaaa 5aaaaaaa n=-4
aaaaaaaa 00000000 n=0
aaaa5555 0000ffff n=0
Edwards-MacBook-Pro:test edwardl$
```