Roll No: 24B-CO-024 Harsh Gaonker

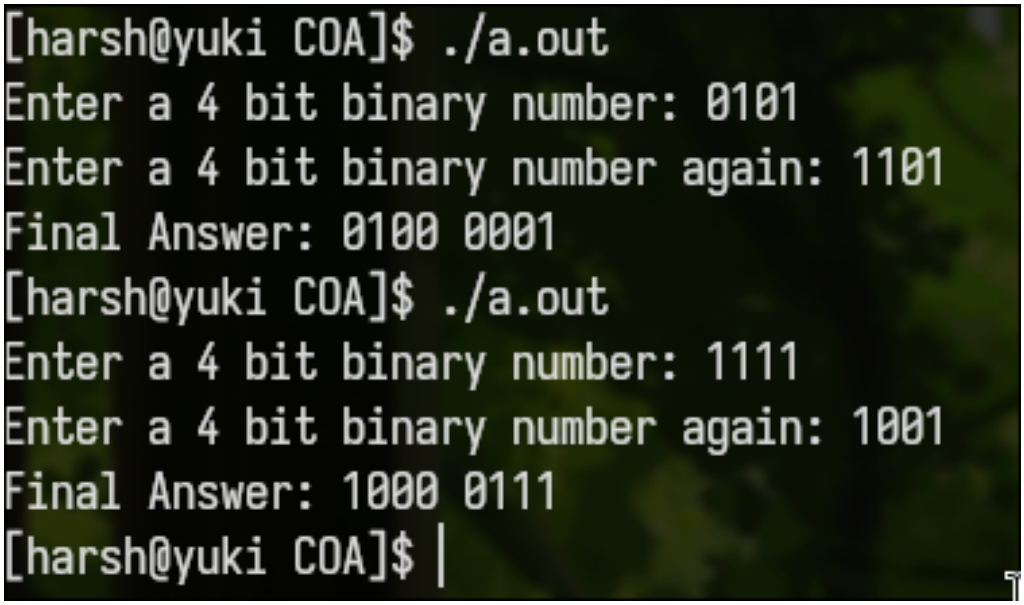
Source Code:

```
#include <stdio.h>
#include <string.h>

typedef struct {
  char carry;
  char A[5];
} add;

add addition(char*, char*);
void shitfRight(add*, char*);

int main(){
  char Q[5];
  char M[5];

  add finalAnswer;
  finalAnswer.carry = '0';
  strcpy(finalAnswer.A, "0000");

  printf("Enter a 4 bit binary number: ");
  scanf("%5s", Q);
  printf("Enter a 4 bit binary number again: ");
  scanf("%5s", M);

  for(int i=0; i<4; i++) {
    if(Q[3] == '1') {
      finalAnswer = addition(finalAnswer.A, M);
      shitfRight(&finalAnswer, Q);
    } else {
      shitfRight(&finalAnswer, Q);
    }
  }

  printf("Final Answer: ");
  printf("%s %s\n", finalAnswer.A, Q);

  return 0;
}

add addition(char *Q, char*M) {
  add sum;
  sum.carry = '0';

  int num;
```

```
  int carry = 0;

  for(int i = 3; i > -1; i--){
    num = carry ^ Q[i]%48 ^ M[i]%48;
    carry = (carry & Q[i]%48) | (carry & M[i]%48) | (Q[i]%48 & M[i]%48);
    sum.A[i] = (char)('0' + num);
  }

  sum.carry = (char)('0' + carry);
  sum.A[4] = '\0';

  return sum;
}

void shitfRight(add* ans, char* Q){
  char carry[2];
  carry[0] = ans->carry;
  carry[1] = '\0';

  char CAQ[25];

  strcpy(CAQ, carry);
  strcat(CAQ, ans->A);
  strcat(CAQ, Q);

  for(int i=8; i>0; i--)
    CAQ[i] = CAQ[i-1];
  CAQ[0] = '0';

  ans->carry = CAQ[0];

  ans->A[0] = CAQ[1];
  ans->A[1] = CAQ[2];
  ans->A[2] = CAQ[3];
  ans->A[3] = CAQ[4];

  Q[0] = CAQ[5];
  Q[1] = CAQ[6];
  Q[2] = CAQ[7];
  Q[3] = CAQ[8];
}
```

Roll No: 24B-CO-024 Harsh Gaonker

Ouput:



Conclusion: C program to perform Multiplication of two unsigned binary numbers was successfully
implemented.