U19CS076 BOOTH’S ASSIGNMENT

```
#include <stdio.h>

#include <math.h>


int a = 0,b = 0, c = 0, a1 = 0, b1 = 0, com[9] = { 1, 0, 0, 0, 0};

int anum[9] = {0}, anumcp[9] = {0}, bnum[9] = {0};

int acomp[9] = {0}, bcomp[9] = {0}, pro[9] = {0}, res[9] = {0};

void binary(){

    int r, r2, i, temp;

    a1 = abs(a);

          b1 = abs(b);

     for (i = 0; i < 9; i++){

          r = a1 % 2;

          a1 = a1 / 2;

          r2 = b1 % 2;

          b1 = b1 / 2;

          anum[i] = r;

          anumcp[i] = r;

          bnum[i] = r2;

          if(r2 == 0){

               bcomp[i] = 1;

          }

          if(r == 0){

               acomp[i] =1;
```

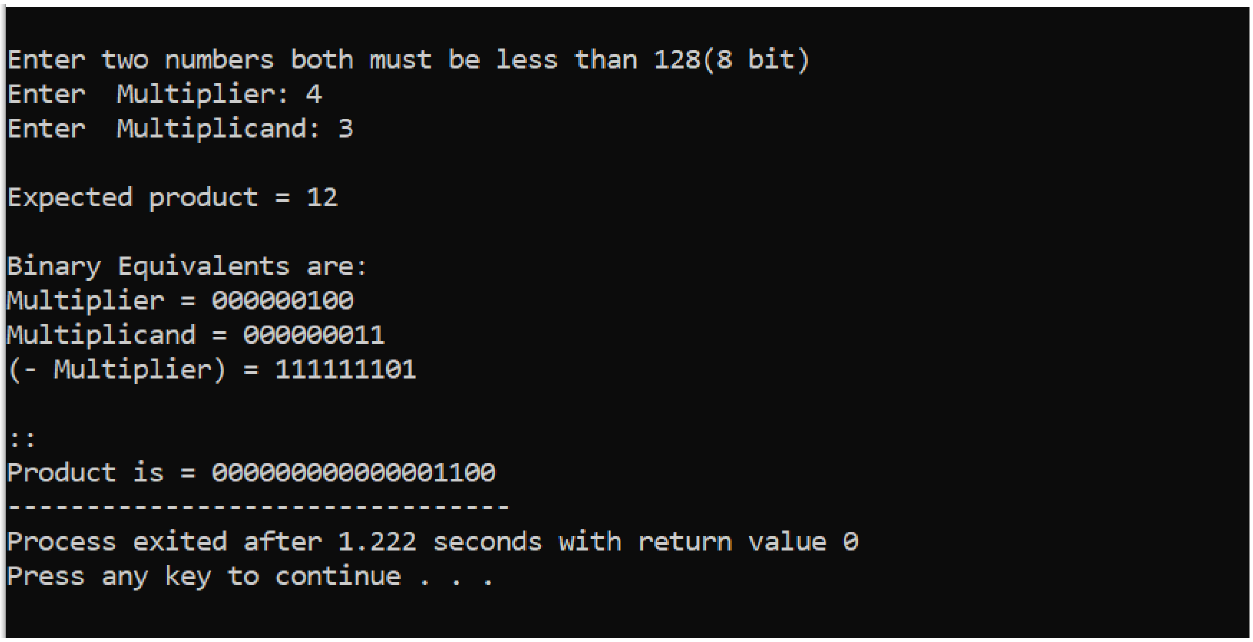
```
        }
  }
c = 0;
for ( i = 0; i < 9; i++){
        res[i] = com[i]+ bcomp[i] + c;
        if(res[i] >= 2){
             c = 1;
        }
        else
             c = 0;
        res[i] = res[i] % 2;
  }
for (i = 8; i >= 0; i--){
  bcomp[i] = res[i];
}
if (a < 0){
   c = 0;
  for (i = 8; i >= 0; i--){
        res[i] = 0;
  }
  for ( i = 0; i < 9; i++){
        res[i] = com[i] + acomp[i] + c;
        if (res[i] >= 2){
             c = 1;
        }
        else
            c = 0;
        res[i] = res[i]%2;
    }
    for (i = 8; i >= 0; i--){
        anum[i] = res[i];
        anumcp[i] = res[i];
    }
  }
  if(b < 0){
    for (i = 0; i < 9; i++){
        temp = bnum[i];
        bnum[i] = bcomp[i];
        bcomp[i] = temp;
    }
  }
}
void add(int num[]){
    int i;
    c = 0;
    for ( i = 0; i < 9; i++){
        res[i] = pro[i] + num[i] + c;
        if (res[i] >= 2){
            c = 1;
        }
```

```
           else{
                c = 0;
           }
           res[i] = res[i]%2;
     }
     for (i = 8; i >= 0; i--){
         pro[i] = res[i];
     }
   printf(":");
   for (i = 8; i >= 0; i--){
     }
}
void rightshift(){
    int temp = pro[8], temp2 = pro[0], i;
    for (i = 1; i < 9  ; i++){
       pro[i-1] = pro[i];
    }
    pro[8] = temp;
    for (i = 1; i < 9  ; i++){
        anumcp[i-1] = anumcp[i];
    }
    anumcp[8] = temp2;


}
```

```
   }

int main(){
  int i, q = 0;


  printf("\nEnter two numbers both must be less than 128(8 bit)");
  do{
      printf("\nEnter  Multiplier: ");
      scanf("%d",&a);
      printf("Enter  Multiplicand: ");
      scanf("%d", &b);
    }while(a >=128 || b >=128);
   printf("\nExpected product = %d", a * b);
   binary();
   printf("\n\nBinary Equivalents are: ");
   printf("\nMultiplier = ");
   for (i = 8; i >= 0; i--){
      printf("%d", anum[i]);
   }
   printf("\nMultiplicand = ");
   for (i = 8; i >= 0; i--){
      printf("%d", bnum[i]);
   }
   printf("\n(- Multiplier) = ");
   for (i = 8; i >= 0; i--){
      printf("%d", bcomp[i]);
   }
```

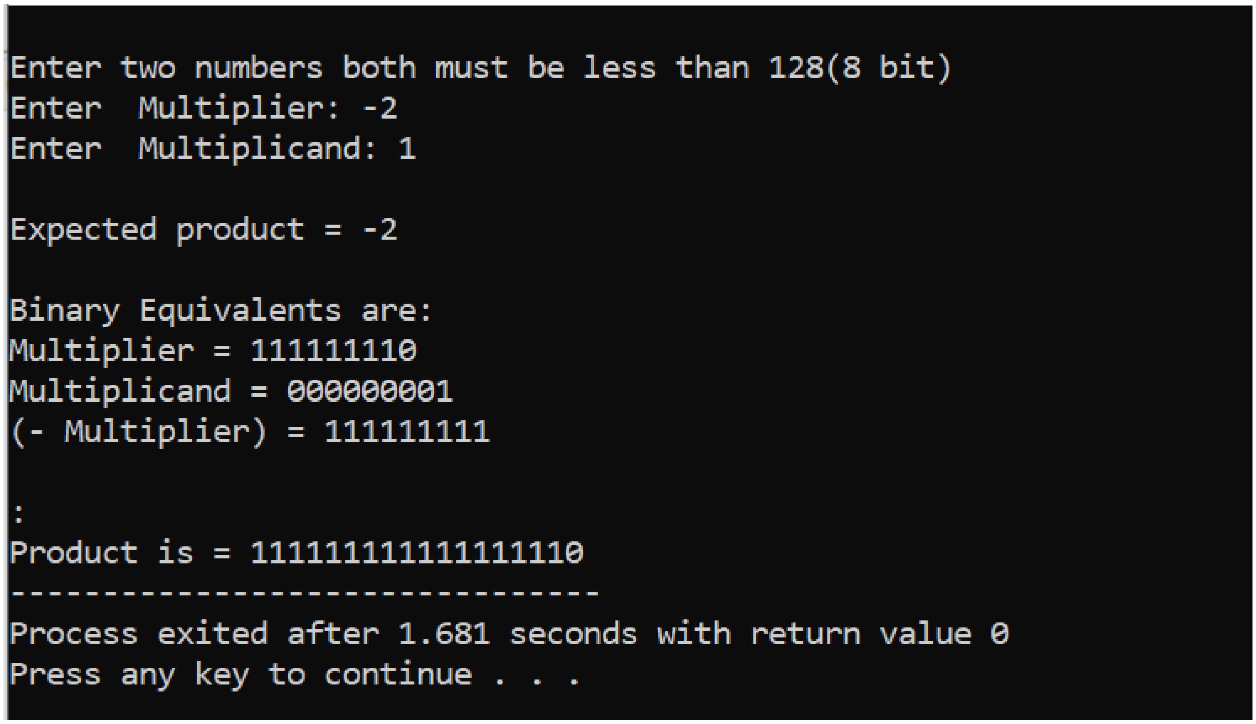
```
    printf("\n\n");

    for (i = 0;i < 9; i++){

          if (anum[i] == q){// shift for 00 or 11

              rightshift();

              q = anum[i];

          }

          else if(anum[i] == 1 && q == 0){

             add(bcomp);

             rightshift();

             q = anum[i];

          }

          else{

             add(bnum);

             rightshift();

             q = anum[i];

          }

     }

     printf("\nProduct is = ");

     for (i = 8; i >= 0; i--){

          printf("%d", pro[i]);

     }

     for (i = 8; i >= 0; i--){

          printf("%d", anumcp[i]);

     }

}
```

A)



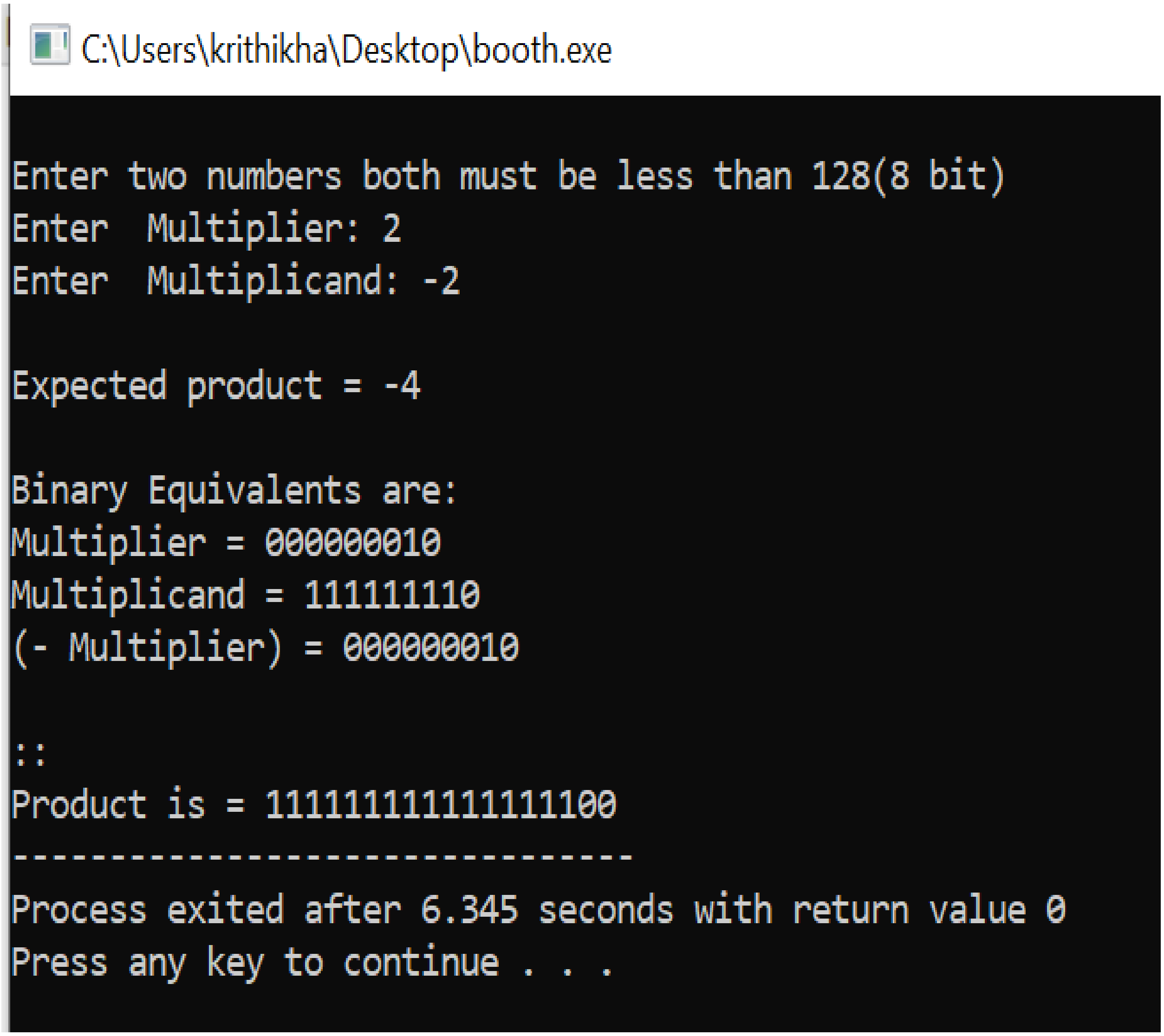
```
Enter two numbers both must be less than 128(8 bit)
Enter  Multiplier: 4
Enter  Multiplicand: 3

Expected product = 12

Binary Equivalents are:
Multiplier = 000000100
Multiplicand = 000000011
(- Multiplier) = 111111101

::
Product is = 000000000000001100
--------------------------------
Process exited after 1.222 seconds with return value 0
Press any key to continue . . .
```

B)



```
Enter two numbers both must be less than 128(8 bit)
Enter  Multiplier: -2
Enter  Multiplicand: 1

Expected product = -2

Binary Equivalents are:
Multiplier = 111111110
Multiplicand = 000000001
(- Multiplier) = 111111111

:
Product is = 111111111111111110
--------------------------------
Process exited after 1.681 seconds with return value 0
Press any key to continue . . .
```

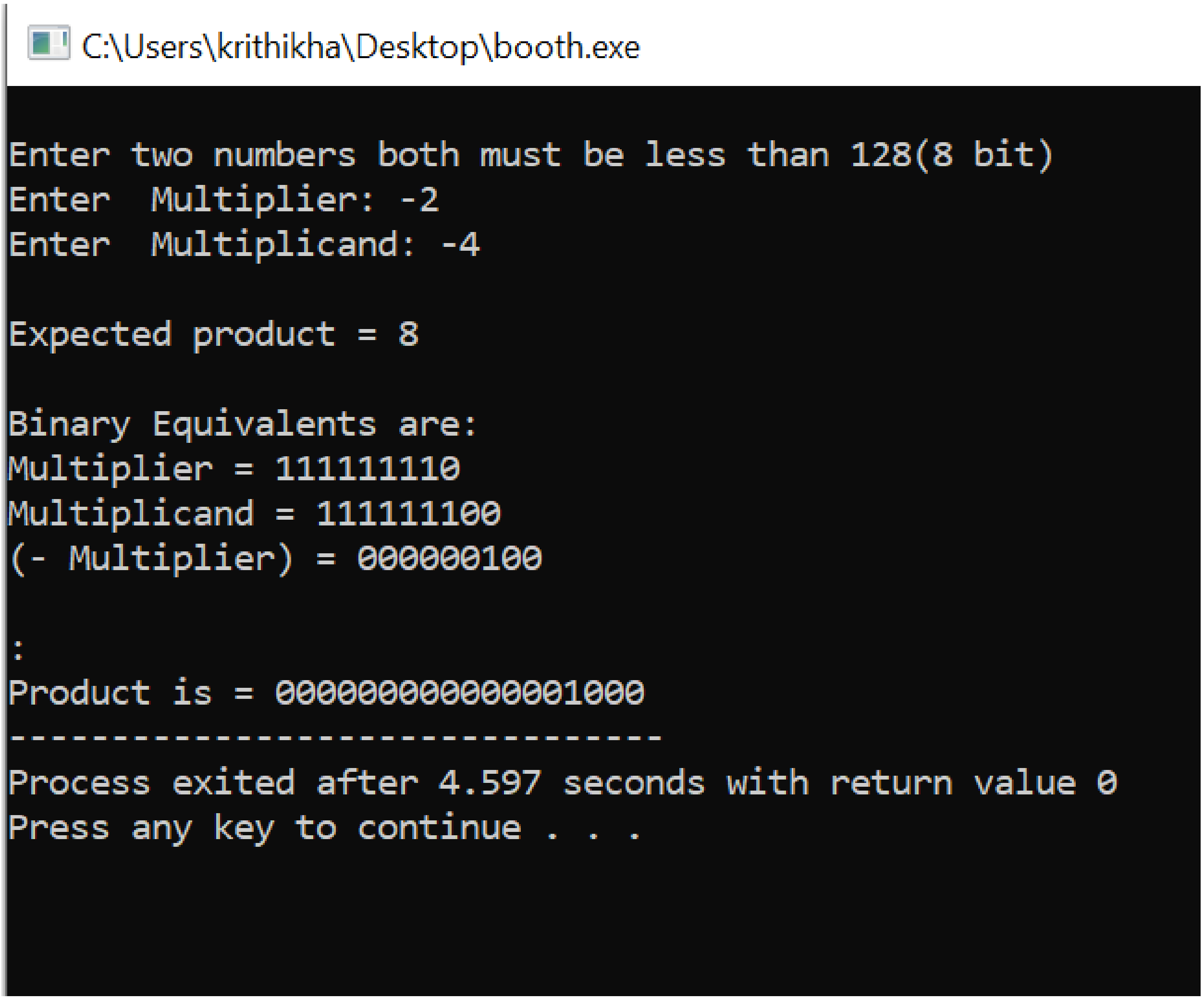
C)



C:\Users\krithikha\Desktop\booth.exe

```
Enter two numbers both must be less than 128(8 bit)
Enter  Multiplier: 2
Enter  Multiplicand: -2

Expected product = -4

Binary Equivalents are:
Multiplier = 000000010
Multiplicand = 111111110
(- Multiplier) = 000000010

::
Product is = 111111111111111100
--------------------------------
Process exited after 6.345 seconds with return value 0
Press any key to continue . . .
```

D)



C:\Users\krithikha\Desktop\booth.exe

```
Enter two numbers both must be less than 128(8 bit)
Enter  Multiplier: -2
Enter  Multiplicand: -4

Expected product = 8

Binary Equivalents are:
Multiplier = 111111110
Multiplicand = 111111100
(- Multiplier) = 000000100

:
Product is = 000000000000001000
--------------------------------
Process exited after 4.597 seconds with return value 0
Press any key to continue . . .
```