EXPERIMENT 17:

Write a program to perform Booth’s multiplication of two signed numbers using any high level language.

PROGRAM :

1. #include <stdio.h>
2. #include <math.h>
4. int a = 0,b = 0, c = 0, a1 = 0, b1 = 0, com[5] = { 1, 0, 0, 0, 0};
5. int anum[5] = {0}, anumcp[5] = {0}, bnum[5] = {0};
6. int acomp[5] = {0}, bcomp[5] = {0}, pro[5] = {0}, res[5] = {0};
8. void binary(){
9. a1 = fabs(a);
10. b1 = fabs(b);
11. int r, r2, i, temp;
12. for (i = 0; i < 5; i++){
13. r = a1 % 2;
14. a1 = a1 / 2;
15. r2 = b1 % 2;
16. b1 = b1 / 2;
17. anum[i] = r;
18. anumcp[i] = r;
19. bnum[i] = r2;
20. if(r2 == 0){
21. bcomp[i] = 1;
22. }
23. if(r == 0){
24. acomp[i] =1;
25. }
26. }
27. *//part for two's complementing*
28. c = 0;
29. for ( i = 0; i < 5; i++){
30. res[i] = com[i]+ bcomp[i] + c;
31. if(res[i] >= 2){
32. c = 1;
33. }
34. else
35. c = 0;
36. res[i] = res[i] % 2;
37. }
38. for (i = 4; i >= 0; i--){
39. bcomp[i] = res[i];
40. }
41. *//in case of negative inputs*
42. if (a < 0){
43. c = 0;
44. for (i = 4; i >= 0; i--){
45. res[i] = 0;
46. }
47. for ( i = 0; i < 5; i++){
48. res[i] = com[i] + acomp[i] + c;
49. if (res[i] >= 2){
50. c = 1;
51. }
52. else
53. c = 0;
54. res[i] = res[i]%2;
55. }
56. for (i = 4; i >= 0; i--){
57. anum[i] = res[i];
58. anumcp[i] = res[i];
59. }
61. }
62. if(b < 0){
63. for (i = 0; i < 5; i++){
64. temp = bnum[i];
65. bnum[i] = bcomp[i];
66. bcomp[i] = temp;
67. }
68. }
69. }
70. void add(int num[]){
71. int i;
72. c = 0;
73. for ( i = 0; i < 5; i++){
74. res[i] = pro[i] + num[i] + c;
75. if (res[i] >= 2){
76. c = 1;
77. }
78. else{
79. c = 0;
80. }
81. res[i] = res[i]%2;
82. }
83. for (i = 4; i >= 0; i--){
84. pro[i] = res[i];
85. printf("%d",pro[i]);
86. }
87. printf(":");
88. for (i = 4; i >= 0; i--){
89. printf("%d", anumcp[i]);
90. }
91. }
92. void arshift(){*//for arithmetic shift right*
93. int temp = pro[4], temp2 = pro[0], i;
94. for (i = 1; i < 5 ; i++){*//shift the MSB of product*
95. pro[i-1] = pro[i];
96. }
97. pro[4] = temp;
98. for (i = 1; i < 5 ; i++){*//shift the LSB of product*
99. anumcp[i-1] = anumcp[i];
100. }
101. anumcp[4] = temp2;
102. printf("**\n**AR-SHIFT: ");*//display together*
103. for (i = 4; i >= 0; i--){
104. printf("%d",pro[i]);
105. }
106. printf(":");
107. for(i = 4; i >= 0; i--){
108. printf("%d", anumcp[i]);
109. }
110. }
112. void main(){
113. int i, q = 0;
114. printf("**\t\t**BOOTH'S MULTIPLICATION ALGORITHM");
115. printf("**\n**Enter two numbers to multiply: ");
116. printf("**\n**Both must be less than 16");
117. *//simulating for two numbers each below 16*
118. do{
119. printf("**\n**Enter A: ");
120. scanf("%d",&a);
121. printf("Enter B: ");
122. scanf("%d", &b);
123. }while(a >=16 || b >=16);
125. printf("**\n**Expected product = %d", a \* b);
126. binary();
127. printf("**\n\n**Binary Equivalents are: ");
128. printf("**\n**A = ");
129. for (i = 4; i >= 0; i--){
130. printf("%d", anum[i]);
131. }
132. printf("**\n**B = ");
133. for (i = 4; i >= 0; i--){
134. printf("%d", bnum[i]);
135. }
136. printf("**\n**B'+ 1 = ");
137. for (i = 4; i >= 0; i--){
138. printf("%d", bcomp[i]);
139. }
140. printf("**\n\n**");
141. for (i = 0;i < 5; i++){
142. if (anum[i] == q){*//just shift for 00 or 11*
143. printf("**\n**-->");
144. arshift();
145. q = anum[i];
146. }
147. else if(anum[i] == 1 && q == 0){*//subtract and shift for 10*
148. printf("**\n**-->");
149. printf("**\n**SUB B: ");
150. add(bcomp);*//add two's complement to implement subtraction*
151. arshift();
152. q = anum[i];
153. }
154. else{*//add ans shift for 01*
155. printf("**\n**-->");
156. printf("**\n**ADD B: ");
157. add(bnum);
158. arshift();
159. q = anum[i];
160. }
161. }
163. printf("**\n**Product is = ");
164. for (i = 4; i >= 0; i--){
165. printf("%d", pro[i]);
166. }
167. for (i = 4; i >= 0; i--){
168. printf("%d", anumcp[i]);
169. }
170. }

OUTPUT: