

```

1. // string multiplication by Shadman
2.
3. #include <iostream>
4. #include <string>
5. #include <sstream>
6. #include <cstdio>
7. #define OVERFLOW 2
8. #define ROW b_len
9. #define COL a_len+b_len+OVERFLOW
10.
11. using namespace std;
12.
13. int getCarry(int num) {
14.     int carry = 0;
15.     if(num>=10) {
16.         while(num!=0) {
17.             carry = num %10;
18.             num = num/10;
19.         }
20.     }
21.     else carry = 0;
22.     return carry;
23. }
24.
25. int num(char a) {
26.     return int(a)-48;
27. }
28.
29. string mult(string a, string b) {
30.     string ret;
31.     int a_len = a.length();
32.     int b_len = b.length();
33.     int mat[ROW][COL];
34.     for(int i =0; i<ROW; ++i) {
35.         for(int j=0; j<COL; ++j) {
36.             mat[i][j] = 0;
37.
38.         }
39.     }
40.
41.     int carry=0, n,x=a_len-1,y=b_len-1;
42.     for(int i=0; i<ROW; ++i) {
43.         x=a_len-1;
44.         carry = 0;
45.         for(int j=(COL-1)-i; j>=0; --j) {
46.             if((x>=0)&&(y>=0)) {
47.                 n = (num(a[x])*num(b[y]))+carry;
48.                 mat[i][j] = n%10;
49.                 carry = getCarry(n);
50.             }
51.             else if((x>=-1)&&(y>=-1)) mat[i][j] = carry;
52.             x=x-1;
53.         }
54.         y=y-1;
55.     }
56.
57.     carry = 0;
58.     int sum_arr[COL];

```

```

59.     for(int i =0; i<COL; ++i) sum_arr[i] = 0;
60.     for(int i=0; i<ROW; ++i) {
61.         for(int j=COL-1; j>=0; --j) {
62.             sum_arr[j] += (mat[i][j]);
63.         }
64.     }
65.     int temp;
66.     for(int i=COL-1; i>=0; --i) {
67.         sum_arr[i] += carry;
68.         temp = sum_arr[i];
69.         sum_arr[i] = sum_arr[i]%10;
70.         carry = getCarry(temp);
71.     }
72.
73.     for(int i=0; i<COL; ++i) {
74.         ret.push_back(char(sum_arr[i]+48));
75.     }
76.
77.     while(ret[0]=='0'){
78.         ret = ret.substr(1,ret.length()-1);
79.     }
80.     return ret;
81. }
82.
83. void printhuge(string a) {
84.
85.     for(string::iterator i = a.begin(); i!=a.end(); ++i) {
86.         cout<<*i;
87.     }
88. }
89. string intToStr(unsigned long long int num) {
90. //takes an int and returns a string
91. stringstream ss;
92. ss << num;
93. return ss.str();
94. }
95.
96. int main() {
97.     unsigned long long int p,q,r,s,T,i,j;
98.
99.     string a,b,c,d;
100.
101.     cin>>T;
102.     for(i=1;i<=T;i++)
103.     {
104.
105.
106.         scanf("%llu",&p);
107.         if(p==0)
108.         {
109.             printf("Case %llu: 0\n",i);
110.             continue;
111.         }
112.         else if(p==1)
113.         {
114.             printf("Case %llu: 0\n",i);
115.             continue;
116.         }

```

```
117.     if(p%2==0)
118.     {
119.         r=p/2;
120.         s=p-1;
121.     }
122.     else
123.     {
124.         r=(p-1)/2;
125.         s=p;
126.     }
127.     a=intToStr(r);
128.
129.     b=intToStr(s);
130.
131.     printf("Case %llu: ",i);
132.
133.     printhuge(mult(a,b));
134.     printf("\n");
135. }
136. return 0;
137. }
```