

# Long Arithmetic

## Code:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <stdbool.h>
4
5 #define N 32
6 #define Base 3 //size of unsigned int array
7
8 unsigned char AAA[N];
9 unsigned char CCC[N];
10 unsigned char DDD[N];
11 unsigned char MAX[N];
12
13 //Initialization
14 void Init(unsigned char Tmp[N],unsigned int n){
15     for(int i=0;i<N;i++){
16         Tmp[i]=n%256;
17         n/=256;
18     }
19 }
20
21 //Copy from B to A
22 void Copy(unsigned char A[N],unsigned char B[N]){
23     for(int i=0;i<N;i++)
24         A[i]=B[i];
25 }
26
27 //Print
28 void Print(unsigned char X[N]){
29     for(int i=N-1;i>=0;i--)
30         printf("%X ",X[i]);
31     printf("\n");
32 }
33
34 //Addition:C=A+B
35 void Add(unsigned char C[N],unsigned char A[N],unsigned char B[N]){
36     int Digits=0;
37     int Res;
38     for(int i=0;i<N;i++){
39         Res=Digits+A[i]+B[i];
40         C[i]=Res%256;
41         Digits=Res/256;
42     }
43 }
44
45 //Multiplication of digit:C=A*B
46 void MulDigit(unsigned char C[N],unsigned char A[N],unsigned char B){
47     Init(C,0);
48     int Digits=0;
49     int Res;
50     for(int i=0;i<N;i++){
51         Res=Digits+A[i]*B;
52         C[i]=Res%256;
53         Digits=Res/256;
54     }
55 }
56
57 //Digit shift forward k position
58 void Shift(unsigned char A[N],int k){
59     for(int i=N-1;i>=k;i--)
60         A[i]=A[i-k];
61     for(int j=k-1;j>=0;j--)
62         A[j]=0;
63 }
```

```

64
65 //Multiplication:C=A+B
66 void Multi(unsigned char C[N],unsigned char A[N],unsigned char B[N]){
67     Init(C,0);
68     for(int i=0;i<N;i++){
69         unsigned char Tmp[N];
70         MulDigit(Tmp,A,B[i]);
71         Shift(Tmp,i);
72         Add(C,C,Tmp);
73     }
74 }
75
76 //The Signal number
77 void Max(unsigned char MAX[N]){
78     for(int i=0;i<=N-2;i++)
79         MAX[i]=255;
80     MAX[N-1]=63;//63*4=252<256 for 3^N
81 }
82
83 //A<B ->true; A>B ->false
84 bool Less(unsigned char A[N],unsigned char B[N]){
85     for(int i=N-1;i>=0;i--){
86         if(A[i]<B[i]) return true;
87         if(A[i]>B[i]) return false;
88     }
89     return true;
90 }
91
92 void Number(){
93     Init(AAA,Base);
94     Init(CCC,1);
95     Print(AAA);
96     Print(CCC);
97     for(int i=1;;i++){
98         printf("%d ",i);
99         Multi(DDD,CCC,AAA);
100        Copy(CCC,DDD);
101        if(!Less(CCC,MAX)){
102            printf("Bigger ");
103            Print(CCC);
104            break;
105        }
106        else
107            printf("Smaller ");
108            Print(CCC);
109    }
110 }
111
112 int main(){
113     Max(MAX);
114     Print(MAX);
115     Number();
116     return 0;
117 }
118
119
120

```

## Result:

$3^N$ :

**Max=[63][255][255][255].....[255]**

```

157 Smaller 1 C9 F8 97 27 24 64 66 3A D3 BE 29 9F C4 94 0 C4 55 13 CA 8A 9F EF 87 A3 C9 A9 9D 64 D 8B 93
158 Smaller 5 5D E9 C5 75 6D 2D 32 B0 7B 3A 7C DF 4D BC 2 4C FF 3B 5F 9F DF CE 96 EB 5C FC D8 2C 28 A2 B9
159 Smaller 10 19 BD 50 60 47 87 98 11 71 AF 76 9D E9 34 6 E6 FD B2 1E DF 9F 6B C4 C2 16 F6 88 84 79 E8 2B
160 Smaller 30 4D 37 F1 20 D6 96 C8 34 55 E 63 D9 BB 9C 14 B4 F9 16 5C 9E DE 43 4E 46 44 E3 99 8D 6D B8 81
161 Bigger 90 E7 A7 D3 62 83 C4 58 9C FF 2B 2B 8D 32 D4 3E 1E EB 43 15 DC 9A C9 EA D2 CE AA CC A8 49 29 83
-----
```

Process exited after 0.5829 seconds with return value 0

$5^N$ :

**Max=[50][255][255][255].....[255]**

```

107 Smaller 1 5C CF E3 D3 5D 80 E3 F D9 B0 B2 D 1 47 54 22 F2 12 AA E6 C6 2 F4 86 33 31 8E 50 8A 13 5D
108 Smaller 6 D0 F 73 20 D3 84 6F 4F 40 73 7A 41 6 64 A4 AE BA 5D 56 81 DE E C6 9E FF F7 C7 92 B2 60 D1
109 Smaller 22 10 4D 3F A4 21 96 2C 8C 42 41 63 45 1F F7 37 69 A3 D2 B0 89 56 49 E1 1A FF D6 E5 DD 7B E4 15
110 Bigger AA 51 82 3E 34 A7 EE DE BD 4B 46 F0 59 9F D4 15 10 33 1D 72 AE AF 71 65 86 FF 32 7D 53 6B 74 69
-----
```

Process exited after 0.5137 seconds with return value 0

$7^N$ :

**Max=[35][255][255][255].....[255]**

```

88 Smaller 0 84 42 90 3A 93 3A 1A CF 15 AA 91 82 42 75 EF 7B 3A 63 98 E3 AA F2 C5 72 B9 C6 4F DE 65 8A 41
89 Smaller 3 9D D1 F1 9A 6 96 BB A9 97 A9 FA 8F D1 39 8C 5E 98 B9 2E 39 AC A3 66 23 14 6C 2F 14 C6 C7 C7
90 Smaller 19 50 BD 9B 36 2E 1F 21 A3 25 A5 D9 EE B8 92 D6 96 2D 10 43 93 B8 77 CA F5 8E F5 49 91 6F 76 71
91 Bigger B1 35 2F 3E 7B 42 D9 EB 76 7 88 F5 87 C 3 DE 1B 3B 71 D9 A B 46 8C B6 E8 B5 2 FA C 3D 17
-----
```

Process exited after 0.3719 seconds with return value 0

$11^N$ :

**Max=[22][255][255][255].....[255]**

```

70 Smaller 0 4 78 45 6D 4D 8F 40 D9 6D E7 27 D1 3F 81 5E BD D6 4C 53 C3 1A BD 84 13 A2 2F C7 26 DC 8B 29
71 Smaller 0 31 2A FB B2 55 27 C9 57 B8 EE B5 FD BA 8F 12 28 35 47 99 62 26 24 AC D7 F8 D 8E AB 79 FA C3
72 Smaller 2 1C D8 D0 A9 A8 B5 A6 C4 F2 41 D1 E7 4 25 C7 BA 4A 13 97 37 A3 93 6D 47 A8 95 21 5E 3D C6 61
73 Bigger 17 3D 50 F7 4A 3F CE 2A 76 68 D4 4 ED 2D 9F 95 1 2E D7 7F 64 7 55 B2 14 3E 68 6F C A7 86 2B
-----
```

Process exited after 0.3543 seconds with return value 0