#include <iostream>

#include<cmath>

#include<cstdio>

#include<iostream>

#include<string>

#include<algorithm>

#define MAX 10002

using namespace std;

#define MAX 1000000001

#define arr\_len 6000

int store[arr\_len] = { 0 };

int BS(int a[],int num, int key) {

int front = 0, back = num - 1;

while (1)

{

int mid = (front + back) / 2;

if (a[mid] >= key&&a[mid-1]<key)

return mid;

else

{

if (a[mid] < key)

front = mid;

else

back = mid;

continue;

}

}

}

int main() {

int num = 0;

for (int a = 0; pow(2, a) < MAX; a++)

{

for (int b = 0; pow(3, b) < MAX; b++)

for (int c = 0; pow(5, c) < MAX; c++)

for (int d = 0; pow(7, d) < MAX; d++)

{

int temp = pow(2, a)\*pow(3, b)\*pow(5, c)\*pow(7, d);

if(temp<MAX&&temp>0)

store[num++] = temp;

}

}

sort(store, store+arr\_len);

//for (int i = 0; i < arr\_len; i++)

//{

// cout << store[i] << endl;

//}

int num2;

scanf("%d",&num2);

for (int i = 0; i < num2; i++)

{

int temp;

scanf("%d", &temp);

int p = lower\_bound(store, store+arr\_len, temp) - store;

printf("%d\n", store[p]);

}

}

#include <iostream>

#include<cmath>

#include<cstdio>

#include<iostream>

#include<string>

#include<algorithm>

#define MAX 10002

using namespace std;

#define MAX 1000000001

#define arr\_len 6000

int store[arr\_len] = { 0 };

int BS(int a[],int num, int key) {

int front = 0, back = num - 1;

while (1)

{

int mid = (front + back) / 2;

if (a[mid] >= key&&a[mid-1]<key)

return mid;

else

{

if (a[mid] < key)

front = mid;

else

back = mid;

continue;

}

}

}

int main() {

int num = 0;

for (int a = 0; pow(2, a) < MAX; a++)

{

for (int b = 0; pow(3, b) < MAX; b++)

for (int c = 0; pow(5, c) < MAX; c++)

for (int d = 0; pow(7, d) < MAX; d++)

{

int temp = pow(2, a)\*pow(3, b)\*pow(5, c)\*pow(7, d);

if(temp<MAX&&temp>0)

store[num++] = temp;

}

}

sort(store, store+arr\_len);

//for (int i = 0; i < arr\_len; i++)

//{

// cout << store[i] << endl;

//}

int num2;

scanf("%d",&num2);

for (int i = 0; i < num2; i++)

{

int temp;

scanf("%d", &temp);

printf("%d\n", store[BS(store, arr\_len, temp)]);

}

}