**Prime Numbers Again**

Attempted by: **2667**

/

Accuracy: **45%**

/

Maximum Points: **20**

/

36 Votes

/

**Share**

Ad-Hoc, Dynamic Programming

**PROBLEM**

**EDITORIAL**

**MY SUBMISSIONS**

**ANALYTICS**

[**DISCUSSIONS**](https://www.hackerearth.com/practice/algorithms/dynamic-programming/introduction-to-dynamic-programming-1/practice-problems/algorithm/prime-numbers-again/discussion/)NEW

Panda can do any problem anytime and anywhere. Panda is doing an extensive research on prime numbers. Milinda has got a question for Panda. The only way for Panda to impress Milinda is by solving this question.  
Given a number ***N***, find the minimum number of primatic numbers which sum upto ***N***.  
A primatic number refers to a number which is either a prime number or can be expressed as power of prime number to itself i.e. primeprime e.g. 4, 27, etc.  
Note: 8, 32, etc are not primatic numbers.  
Panda is very sad since he is unable to solve the problem. Please help Panda in solving this problem.



**Input Format:**  
The first line will contain two integers: ***T***, the number of test cases.  
Each test case consists of a single integer ***N***.

**Output Format:**  
For each query output the minimum number of primatic numbers which can sum upto ***N***.

**Constraints:**  
***1*** <= ***T*** <= ***105***  
***2*** <= ***N*** <= ***104***

**Subtask 1:**  
***T*** = ***100***, ***2*** <= ***N*** <= ***1000*** - 20 points

**Subtask 2:**  
***T*** = ***105***, ***2*** <= ***N*** <= ***104*** - 80 points

**SAMPLE INPUT**

2

6

3

**SAMPLE OUTPUT**

2

1

**Explanation**

1. The number 6 can be represented as 2 + 2 + 2, 3 + 3, etc. The smallest one among these is 3 + 3 consisting of 2 primatic numbers.
2. The number 3 is itself primatic. Hence the answer is 1.
3. #include<bits/stdc++.h>
4. #define pb push\_back
5. #define int long long int
6. using namespace std;
7. vector<int> primatic;
8. bool isprime[10001];
9. void seive()
10. {
11. for(int i=2;i<10000;i++)
12. {
13. if(isprime[i])
14. {
15. for(int j=i\*i;j<=10000;j+=i)
16. isprime[j]=false;
17. }
18. }
19. isprime[0]=isprime[1]=false;
20. isprime[4]=isprime[27]=isprime[3125]=true;
21. for(int i=2;i<10001;i++)
22. if(isprime[i])
23. primatic.pb(i);
24. }
25. main()
26. {
27. int t;
28. cin>>t;
29. memset(isprime,true,sizeof(isprime));
30. seive();
31. sort(primatic.begin(),primatic.end());
32. int dp[10001]={0};
33. for(int i=2;i<10001;i++)
34. if(isprime[i])
35. dp[i]=1;
36. for(int i=4;i<10001;i++)
37. {
38. if(isprime[i])
39. continue;
40. int ans=10000000000;
41. for(int j=0;j<primatic.size() && i>=primatic[j];j++)
42. {
43. if(i-primatic[j]!=1)
44. ans=min(ans,1+dp[i-primatic[j]]);
45. }
46. dp[i]=ans;
47. }
48. while(t--)
49. {
50. int n;
51. cin>>n;
52. cout<<dp[n]<<"\n";
53. }
54. }