**How to design a tiny URL or URL shortener?**

[3.6](https://www.geeksforgeeks.org/medium/)

How to design a system that takes big URLs like “https://www.geeksforgeeks.org/count-sum-of-digits-in-numbers-from-1-to-n/” and converts them into a short 6 character URL. It is given that URLs are stored in database and every URL has an associated integer id.

One important thing to note is, the long url should also be uniquely identifiable from short url. So we need a [Bijective Function](https://en.wikipedia.org/wiki/Bijection)

[**We strongly recommend that you click here and practice it, before moving on to the solution.**](https://practice.geeksforgeeks.org/problem-page.php?pid=495)

One **Simple Solution** could be Hashing. Use a hash function to convert long string to short string. In hashing, that may be collisions (2 long urls map to same short url) and we need a unique short url for every long url so that we can access long url back.

A **Better Solution** is to use the integer id stored in database and convert the integer to character string that is at most 6 characters long. This problem can basically seen as a base conversion problem where we have a 10 digit input number and we want to convert it into a 6 character long string.

Below is one important observation about possible characters in URL.

A URL character can be one of the following  
1) A lower case alphabet [‘a’ to ‘z’], total 26 characters  
2) An upper case alphabet [‘A’ to ‘Z’], total 26 characters  
3) A digit [‘0’ to ‘9’], total 10 characters

There are total 26 + 26 + 10 = 62 possible characters.

So the task is to convert a decimal number to base 62 number.

To get the original long url, we need to get url id in database. The id can be obtained using base 62 to decimal conversion.

Below is a C++ program based on this idea.

|  |
| --- |
| // C++ prgram to generate short url from intger id and  // integer id back from short url.  #include<iostream>  #include<algorithm>  #include<string>  using namespace std;    // Function to generate a short url from intger ID  string idToShortURL(long int n)  {      // Map to store 62 possible characters      char map[] = "abcdefghijklmnopqrstuvwxyzABCDEF"                   "GHIJKLMNOPQRSTUVWXYZ0123456789";        string shorturl;        // Convert given integer id to a base 62 number      while (n)      {          // use above map to store actual character          // in short url          shorturl.push\_back(map[n%62]);          n = n/62;      }        // Reverse shortURL to complete base conversion      reverse(shorturl.begin(), shorturl.end());        return shorturl;  }    // Function to get integer ID back from a short url  long int shortURLtoID(string shortURL)  {      long int id = 0; // initialize result        // A simple base conversion logic      for (int i=0; i < shortURL.length(); i++)      {          if ('a' <= shortURL[i] && shortURL[i] <= 'z')            id = id\*62 + shortURL[i] - 'a';          if ('A' <= shortURL[i] && shortURL[i] <= 'Z')            id = id\*62 + shortURL[i] - 'A' + 26;          if ('0' <= shortURL[i] && shortURL[i] <= '9')            id = id\*62 + shortURL[i] - '0' + 52;      }      return id;  }    // Driver program to test above function  int main()  {      int n = 12345;      string shorturl = idToShortURL(n);      cout << "Generated short url is " << shorturl << endl;      cout << "Id from url is " << shortURLtoID(shorturl);      return 0;  } |

Output:

Generated short url is dnh

Id from url is 12345

Optimization: We can avoid reverse step in idToShortURL(). To make sure that we get same ID back, we also need to change shortURLtoID() to process characters from end instead of beginning.