```c
#include <cs50.h>
#include <stdio.h>
#include <math.h>

int main(void)
{
    // Number of digits
    const int N = 16;
    long numbers = 0;
    do
    {
        numbers = get_long("Number: ");
    }
    while (numbers < 10000);

    // Get digits
    int digits[N];
    long numbers1 = numbers;
    for (int i = 0; i < N; i++)
    {
        // printf("%li\n", numbers1);
        digits[i] = numbers1 / (long) pow(10, N - 1 - i);
        numbers1 -= digits[i] * (long) pow(10, N - 1 - i);
    }

    // Print out digits
    // for (int i = 0; i < N; i++)
    // {
    //     printf("%f", pow(10, N - 1 -i));
    //     printf("%i   %i\n",i, digits[i]);
    // }

    // Whether it is valid
    // Luhn's Algorithm check
    // First non zero value check

    // Luhn's algorithm, sum of multiplied
    int sum_multiplied = 0;
    int sum_other = 0;
    int sum_up = 0;
    int sum_product_digits = 0;
    for (int i = N - 1; i >= 0 ; i -= 2)
    {
        sum_product_digits = ((digits[i - 1] * 2) % 10) + (int)((digits[i - 1] * 2) / 10);
```

```
        sum_multiplied += sum_product_digits;
        sum_other += digits[i];
    }

    sum_up = sum_multiplied + sum_other;

    // debug
    // printf("The remainder is %i\n",sum_up % 10);
    // printf("The sum is %i  %i %i\n",sum_multiplied, sum_other, sum_up);

    // Luhn's algorithm

    if ((sum_up % 10) == 0)
    {
        // Check AMEX
        if (((digits[0] == 0) & (digits[1] == 3) & digits[2] == 4) | ((digits[0] == 0) & (digits[1] == 3) & (digits[2]
== 7)))
        {
            printf("AMEX\n");
            return 0;
        }

        // Check MASTERCARD
        if ((digits[0] == 5) & ((digits[1] >= 1) & (digits[1] <= 5)))
        {
            printf("MASTERCARD\n");
            return 0;
        }

        // Check VISA
        if ((digits[0] == 4) | ((digits[0] == 0) & (digits[1] == 0) & (digits[2] == 0) & (digits[3] == 4)))
        {
            printf("VISA\n");
            return 0;
        }
    }
    // Else
    printf("INVALID\n");
    return 0;
}
```