rExtEvenDigits

Write a **recursive** function r**ExtEvenDigits()** that extracts the even digits from a positive number *num*, and combines the even digits sequentially into a new number. If the input number *num* does not contain any even digits, then returns -1. The function passes the result to the caller through the pointer parameter *evenPtr*. For example, if *num* is 1234, then 24 is returned through *evenPtr*; and if *num* is 1357, then –1 is returned. You may assume that the input number does not contain the digit 0. The function prototype is given below:

```
void rExtEvenDigits(int num, int *evenPtr);
```

Write a C program to test the function.

A sample program to test the function is given below:

```
#include <stdio.h>
void rExtEvenDigits(int num, int *evenPtr);
int main()
{
   int number, result;

   printf("Enter a number: \n");
   scanf("%d", &number);
   rExtEvenDigits(number, &result);
   printf("rExtEveneDigits(): %d\n", result);
   return 0;
}
void rExtEvenDigits(int num, int *evenPtr)
{
   /* Write your code here */
}
```

Some sample input and output sessions are given below:

```
(1) Test Case 1:
    Enter a number:
    1234
    rExtEvenDigits(): 24

(2) Test Case 2:
    Enter a number:
    1357
    rExtEvenDigits(): -1

(3) Test Case 3:
    Enter a number:
    2468
    rExtEvenDigits(): 2468

(4) Test Case 4:
    Enter a number:
    6
    rExtEvenDigits(): 6
```