Aim:

To reverse a 32-bit signed integer and ensure the result is also within the 32-bit signed integer range.

Algorithm:

1. Read an integer.
2. Initialize reversed number to 0.
3. Extract each digit and append to reversed number.
4. Check for overflow.
5. Return reversed number.

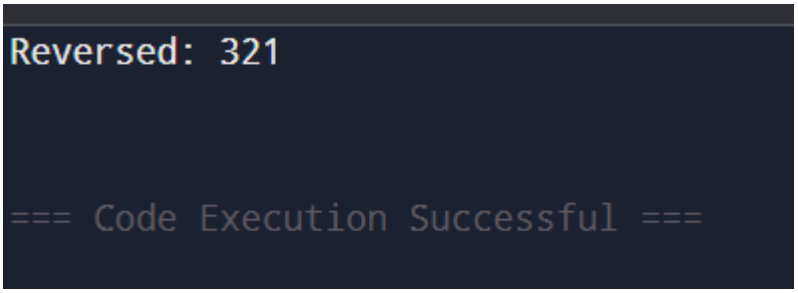
C Code:

```
#include <stdio.h>
#include <limits.h>

int reverse(int x) {
    int rev = 0;
    while (x != 0) {
        int pop = x % 10;
        x /= 10;
        if (rev > INT_MAX/10 || (rev == INT_MAX / 10 && pop > 7))
return 0;
        if (rev < INT_MIN/10 || (rev == INT_MIN / 10 && pop < -8))
return 0;
        rev = rev * 10 + pop;
    }
    return rev;
}

int main() {
    int num;
    printf("Enter a 32-bit integer: ");
    scanf("%d", &num);
    printf("Reversed integer: %d\n", reverse(num));
    return 0;
}
```

Input:

1234

Output:



Result:

The program correctly reverses the input integer.