

```
#include <stdio.h>
int NumofDigits(int);

int main(){

    int n , result;

    printf("Please enter a positive number: ");
    scanf("%d",&n);

    result = NumofDigits(n);
    printf("Entered number has %d digits.",result);

    return 0;
}

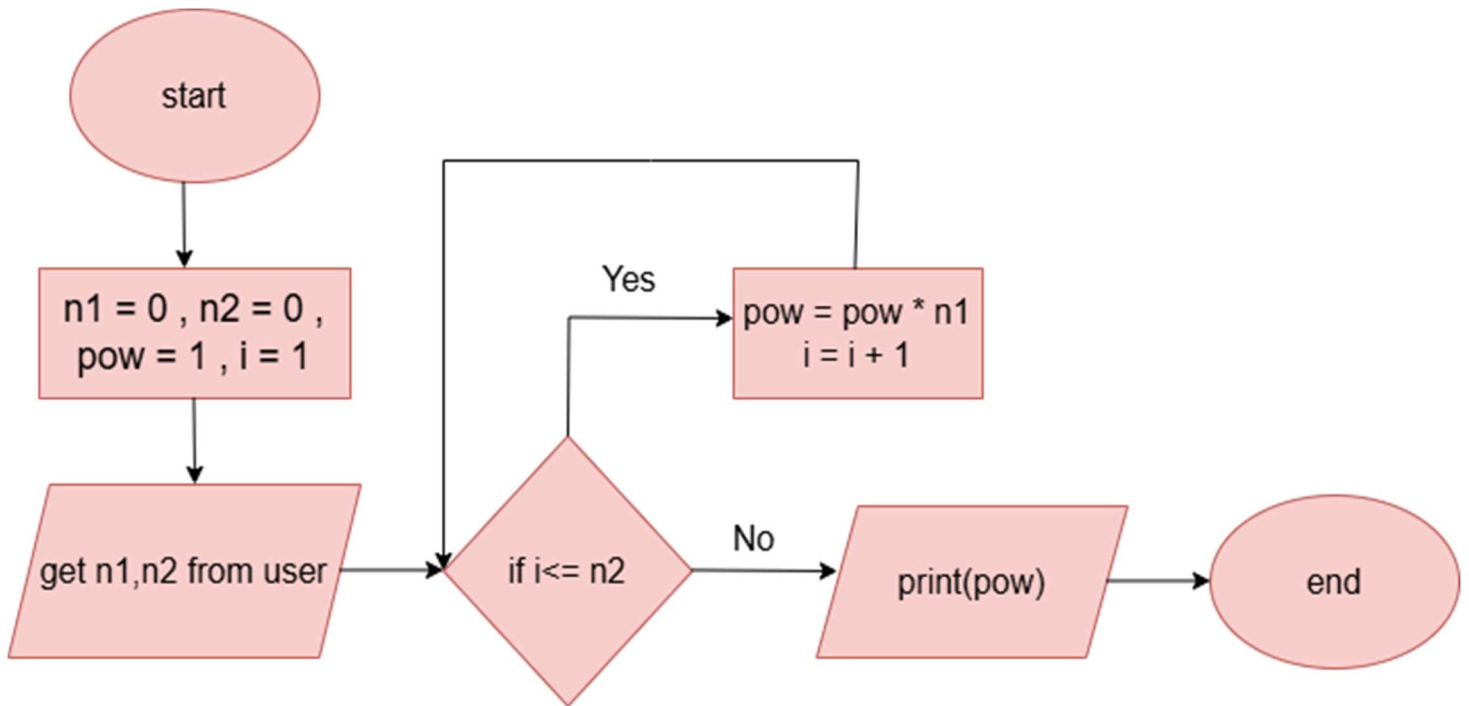
int NumofDigits(int n){

    int digits = 1;

    while ( n/10 != 0){

        digits ++;
        n = n/10;
    }

    return digits;
}
```



```
#include <stdio.h>
int PowerCalculator(int , int);

int main(){

    int base , exp;

    printf("Please enter base: ");
    scanf("%d",&base);

    printf("Please enter exponent: ");
    scanf("%d",&exp);

    int result = PowerCalculator(base , exp);

    printf("%d power by %d is equal to %d",base,exp,result);

    return 0;
}

int PowerCalculator(int base , int exp){

    int i=1 , pow = 1 ;

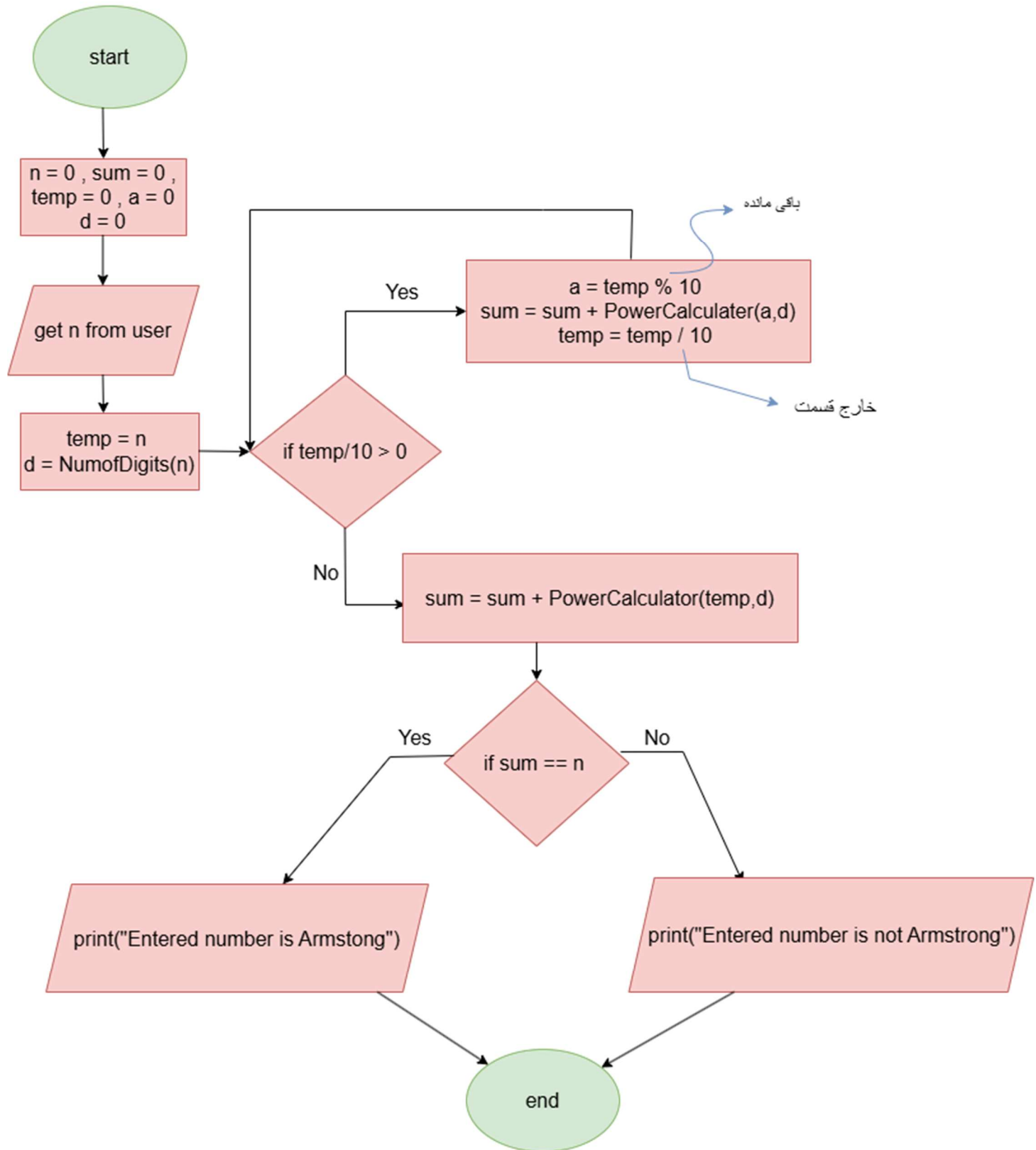
    for( i = 1 ; i <= exp ; i++){

        pow *= base ;
    }

    return pow;
}
```

Heading

NumofDigits and PowerCalculator functions are defined in last algorithms .



```
#include <stdio.h>
int NumofDigits(int);
int PowerCalculator(int , int);
int isArmstrong(int);

int main(){

    int n;
    printf("Please enter an integer: ");
    scanf("%d",&n);
    printf("Result is %d",isArmstrong(n));
}

int isArmstrong(int n){

    int temp = n;
    int d = NumofDigits(n);
    int a , sum = 0;

    while ( temp / 10 > 0 ){

        a = temp % 10;
        sum += PowerCalculator(a,d);
        temp = temp / 10 ;
    }

    sum += PowerCalculator(temp,d);

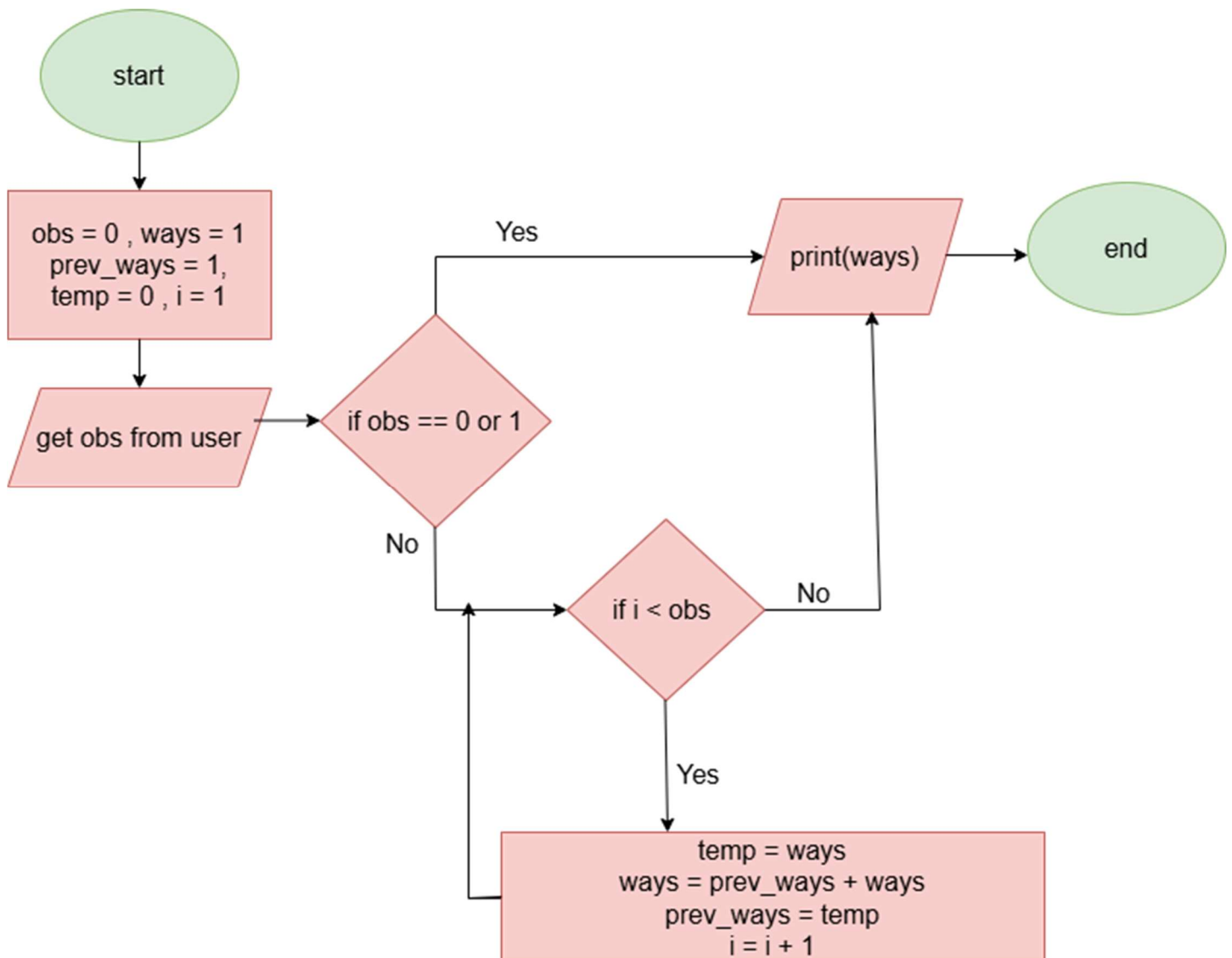
    if (sum == n){
        return 1;
    }
    else{
        return 0;
    }
}

// Will be continued in the next page
```

```
int NumofDigits(int x){  
    int digits = 1;  
    while ( x/10 != 0){  
        digits ++;  
        x = x/10;  
    }  
    return digits;  
}  
  
int PowerCalculator(int base , int exp){  
    int i=1 , pow = 1 ;  
    for( i = 1 ; i <= exp ; i++){  
        pow *= base ;  
    }  
    return pow;  
}
```

- (1) تست و دیباگ توابع void به دلیل عدم وجود یک مقدار بازگشتی مشخص اندکی با مشکل همراه است . در این توابع تشخیص خطا و اطلاع آن به کاربر می تواند دشوار باشد.
- (2) اگر تابعی کارهای متعددی انجام دهد و هیچ مقداری را برنگرداند ، درک نقش آن در برنامه دشوار می شود . برای مثال اگر تابعی محاسبات پیچیده ای انجام می دهد و نتیجه آن برای ادامه برنامه ضروری است ، استفاده از void باعث می شود که این نتیجه از دست برود.

الف) (مشابه سازی با سری فیبوناچی)




```
#include <stdio.h>
int NumberofWays(int);
// Somehow similar to fibo series

int main(){

    int obs;

    printf("Please enter number of obstacles: ");
    scanf("%d",&obs);

    printf("Number of ways: %d",NumberofWays(obs));
    return 0;
}

int NumberofWays(int n){

    int temp ;
    int prev_ways = 1 , ways = 1;

    if ( (n == 0) || (n == 1) ) {

        return 1;
    }

    else {

        for ( int i=1 ; i < n ; i++){

            temp = ways ;
            ways = ways + prev_ways ;
            prev_ways = temp ;
        }
        return ways;
    }
}
```