

## divide

Write a function that performs integer division by subtraction. The function computes the quotient and remainder of dividing  $m$  by  $n$ . Both  $m$  and  $n$  are positive integers. Write the function in two versions, and the function prototypes are given below:

```
int divide1(int m, int n, int *r);
void divide2(int m, int n, int *q, int *r);
```

In the first version **divide1()**, the function returns the quotient of dividing  $m$  by  $n$ , and the remainder is passed to the caller through the pointer parameter  $r$ . In the second version **divide2()**, the pointer variable  $q$  is used to store the quotient which will be returned to the caller, and the remainder is passed to the caller through the pointer parameter  $r$ . Please note that in this question, you are not allowed to use the division (/) and modulus (%) operators.

A sample program template is given below to test the functions:

```
#include <stdio.h>
int divide1(int m, int n, int *r);
void divide2(int m, int n, int *q, int *r);
int main()
{
    int m, n, q, r;

    printf("Enter two numbers (m and n): \n");
    scanf("%d %d", &m, &n);
    q = divide1(m, n, &r);
    printf("divide1(): quotient %d remainder %d\n", q, r);
    divide2(m, n, &q, &r);
    printf("divide2(): quotient %d remainder %d\n", q, r);
    return 0;
}
int divide1(int m, int n, int *r)
{
    /* Write your code here */
}
void divide2(int m, int n, int *q, int *r)
{
    /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:

Enter two numbers (m and n):

10 3

divide1(): quotient 3 remainder 1

divide2(): quotient 3 remainder 1

(2) Test Case 2:

Enter two numbers (m and n):

3 5

divide1(): quotient 0 remainder 3

divide2(): quotient 0 remainder 3

(3) Test Case 3:

Enter two numbers (m and n):

32 7

divide1(): quotient 4 remainder 4

divide2(): quotient 4 remainder 4

(4) Test Case 4:

Enter two numbers (m and n):

1 7

divide1(): quotient 0 remainder 1

divide2(): quotient 0 remainder 1

```
int divide1(int m, int n, int *r)
{
    int count=0;

    while(m>0)
    {
        m=m-n;
        count++;
    }
    if(m<0)
    {
        m=m+n;
    }
    *r = m;
    if (count!=0)
    {count--;}
    return count;
}

void divide2(int m, int n, int *q, int *r)
{
    int count=0;

    while(m>0)
    {
        m=m-n;
        count++;
    }
    if(m<0)
    {
        m=m+n;
    }
    *r = m;
    if (count!=0)
    {count--;}
    *q = count;
}
```