gcd – greatest common divisor

Write a C function that computes the greatest common divisor of two positive numbers. For example, if num1 is 4 and num2 is 7, then the function will return 1; if num1 is 4 and num2 is 32, then the function will return 4; and if num1 is 4 and num2 is 38, then the function will return 2. Write two iterative versions of the function. The function **gcd11()** returns the computed result, while **gcd2()** passes the result through the pointer parameter result. The function prototypes are given as follows:

```
int gcd1(int num1, int num2);
void gcd2(int num1, int num2, int *result);
```

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

```
#include <stdio.h>
int gcd1(int num1, int num2);
void gcd2(int num1, int num2, int *result);
int main()
{
  int x,y,result=-1;

  printf("Enter 2 numbers: \n");
  scanf("%d %d", &x, &y);
  printf("gcd1(): %d\n", gcd1(x, y));
  gcd2(x,y,&result);
  printf("gcd2(): %d\n", result);
  return 0;
}
int gcd1(int num1, int num2)
{
  /* Write your code here */
}
void gcd2(int num1, int num2, int *result)
{
  /* Write your code here */
}
```

Some sample input and output sessions are given below:

(1) Test Case 1:

```
Enter 2 numbers:
4 7
gcd1(): 1
gcd2(): 1
```

(2) Test Case 2:

```
Enter 2 numbers:
32 4
gcd1(): 4
gcd2(): 4
```

(3) Test Case 3:
Enter 2 numbers:
4 38
gcd1(): 2
gcd2(): 2