```
#include<stdio.h>
int maximum(int num1, int num2);

int main(void)
{
   int n1, n2, max;
   printf("Enter two integers: ");
   scanf("%d %d",&n1,&n2);

   max = maximum(n1, n2);

   printf("Max = %d\n", max);
   return 0;
}

int maximum(int num1, int num2)
{
   int theMax;


   if (num1 > num2)
   {
      theMax = num1;
   }
   else
   {
      theMax = num2;
   }
   return theMax;
}
```

1. Write the prototype for maximum.

2. Write the arguments needed send data to maximum.

3. What is the type of theMax?

```
#include <stdio.h>

int maximum(int num1, int num2);
void displayMax(int n1, int n2, int max);

int main()
{
   int num1=5, num2=8, max;

   max = maximum(num1,num2);
   displayMax(num1, num2, max);

   displayMax(num1, num2, maximum(num1,num2));

   return 0;
}

int maximum(int num1, int num2)
{
   int maxNum;
   if (num1 > num2)
   {
      maxNum = num1;
   }
   else
   {
      maxNum = num2;
   }
   return maxNum;
}

void displayMax(int n1, int n2, int max)
{
   printf("The maximum of %d and %d is %d.\n",n1,n2,max);
}
```

assigning value

use funct value

No Return

```
«---jGRASP exec: H:\My Documents\comp 1200\Hundley\Examples\a.exe

The maximum of 5 and 8 is 8.
The maximum of 5 and 8 is 8.

 ----jGRASP: operation complete.
```

```c
/* call-by-value vs call_by_ref */
#include <stdio.h>

float cal_tax     (float  price, float tax_rate);
void  tax_included(float *price, float tax_rate);

int main()
{
   float price=2.99, tax_rate=0.06, tax;

   tax = cal_tax(price, tax_rate);
   price = price + tax;
   printf("The price including tax = $%4.2f\n",price);

   price = 2.99;
   tax_included(&price, tax_rate);
   printf("The price including tax = $%4.2f\n",price);
   return 0;
}

float cal_tax(float  price, float tax_rate)
{
   return (price * tax_rate);
}

void  tax_included(float *price, float tax_rate)
{
   *price = *price * (1.0 + tax_rate);
}
```

```
----jGRASP exec: H:\My Documents\comp 1200\Hundley\Examples\a.exe

The price including tax = $3.17
The price including tax = $3.17

----jGRASP: operation complete.
```