

Write a program that asks the user to enter two decimal numbers. The program should calculate and display the product and quotient of the two numbers. Use a function `Product()` to calculate the product. The function should have two arguments, which are the two numbers that the user inputs. The function should return the product of the numbers. Use a function `Quotient()` that has two arguments, which are the numbers input by the user. The function should return the quotient of the first number divided by the second. If the second number is zero (recall that division by zero is not allowed), display an error message and exit the program.

---

```
#include <iostream>
using namespace std;

float Product(float,float);

float Quotient(float,float);

int main()
{
    float num1,num2;
    cout << "Enter two decimal numbers: ";
    cin >> num1 >> num2;
    cout << "the product of the numbers: " << Product(num1,num2);
    if(num2==0)
        cout << "\nrecall that division by zero is not allowed. " << endl;
    else
        cout << "\nthe quotient of the first number divided by the second: "
        << Quotient(num1,num2) << endl;
    return 0;
}

float Product(float num1 ,float num2 )
{
    return num1*num2;
}

float Quotient(float num1 ,float num2)
{
    return num1/num2;
}
```

---



Write a program that asks the user to enter two decimal numbers. The program should display the larger of the two numbers. Use a function `Max()` to find the larger number. `Max()` should have two arguments, which are the numbers that the user input. `Max()` should return the larger of the two numbers.

---

```
#include <iostream>
#include <iostream>
using namespace std;
float Max(float,float);
int main()
{
    float num1,num2;
    cout << "Enter two decimal numbers: ";
    cin >> num1 >> num2;
    cout << "\nthe larger number: " << Max(num1,num2) << endl;
    return 0;
}
float Max(float num1 ,float num2 )
{
    if(num1>num2)
        return num1;
    else
        return num2;
}
```

---



Write a program that raises an integer to a positive integer power. The program should prompt the user to enter the base and the exponent, and the program should display the corresponding power. Use a function `Power()` to calculate the power. Pass the base and the exponent to the function. `Power()` should calculate the power by multiplying base by itself exponent times. The function should return the value of the power.

---

```
#include <iostream>
using namespace std;
int Power(int,int);
int main()
{
    int base,exponent;
    cout << "enter the base: ";
    cin >> base;
    cout << "enter the exponent: ";
    cin >> exponent;
    cout << "\nthe value of the power: " << Power(base,exponent) << endl;
    return 0;
}
int Power(int base,int exponent)
{
    int result=1;
    for(int i=0;i<exponent;i++)
        result*=base;
    return result;
}
```

---



The Big Racket Tennis Club wants you to write a program to calculate the yearly dues for its members. The program should prompt the user for the membership type (either 'I' for individual or 'F' for family) and the number of years the person has been a member. Use a function `Get_Member_Type()` that prompts the user for his or her membership type, obtains the type from the user and returns the type to `main()`. Use a function `Years()` that prompts the user for the number of years he or she has been a member and returns that number to `main()`.

\* Use a function `Dues()` to calculate the member's dues. Pass the member type and number of years to the function. Calculate the dues as follows: If the person is a family member and has been a club member for more than three years, the dues are \$2400.00. If the person is a family member and has been a club member for three years or less, the dues are \$3000.00. If the person is an individual member and has been a club member for more than five years, the dues are \$1500.00. If the person is an individual member and has been a club member for five years or less, the dues are \$1900.00. The function `Dues()` should return the dues to `main()`.

\* Use a function `Display_Member_Info()` to display the type of membership, the number of years the person has been a member, and the dues, all of which should be passed to the function as arguments.

---

```
#include <iostream>
using namespace std;
char Get_Member_Type();
int Years();
int Dues(char,int);
void Display_Member_Info(char,int,int);
int main()
{
    char Type = Get_Member_Type();
    int year = Years();
    int dues = Dues(Type,year);
    Display_Member_Info(Type,year,dues);
    return 0;
}
char Get_Member_Type()
{
    char Type;
    cout << "the membership type (either 'I' for individual or 'F' for family): ";
    cin >> Type;
    return Type;
}
int Years()
{
    int year;
    cout << "\nthe number of years the person has been a member: ";
    cin >> year;
    return year;
}

int Dues(char type,int year)
{
    if(type=='f'||type=='F')
    {
        if(year>3)
            return 2400;
        else if(year<=3)
```





```
        return 3000;
    }
    else if(type=='i'||type=='I')
    {
        if(year>5)
            return 1500;
        else if(year<=5)
            return 1900;
        }
    }
```

```
void Display_Member_Info(char type,int year,int dues)
```

```
{
    cout << "\nthe type of membership: " << type;
    cout << "\nthe number of years the person has been a member: " << year;
    cout << "\nthe dues: " << dues;
}
```

---

A piece of equipment, such as a computer, loses value over time. The amount of this lost value is called **depreciation**.

The simplest way to determine depreciation is by the **straight-line** method. If the asset has a useful life of  $n$  years, the straight-line method of depreciation assumes a depreciation of  $1/n$  of the item's value each year. Assume a particular \$3,000 computer has a useful life of five years. At the end of the five years, assume it has a **scrap value** of \$500. Thus, the net cost of the computer is  $3000 - 500 = \$2500$ . Using the straight-line method, the computer depreciates \$500 ( $= 2500 \times 1/5$ ) each of the five years of its useful life. After two years, the computer depreciates \$1,000. Thus, the **book value** of the computer after two years (the difference between the cost of the item and the depreciation to date) is \$2,000 ( $= 3000 - 1000$ ).

Write a program that calculates the depreciation of an asset using the straight-line method. The program should ask the user for the initial cost of the item, the scrap value of the item, and the item's useful life in years. Use a function `Depreciation()` that calculates and displays a table that shows the book value of the item after each year of its useful life. The function should have three arguments the initial cost, the scrap value, and the life of the item. The function should not return a value.

---

```
#include <iostream>
#include <iomanip>
using namespace std;
void Depreciation(int,float,int);
int main()
{
    int cost,years;
    float scrap;
    cout << "the initial cost of the item: ";
    cin >> cost;
    cout << "\nthe scrap value of the item: ";
    cin >> scrap;
    cout << "\nthe item's useful life in years: ";
    cin >> years;
    Depreciation(cost,scrap,years);
    return 0;
}
```



```
void Depreciation(int cost,float scrap,int years)
{
    cout << "\n _____" << endl;
    cout << setw(10) << "years" << setw(10) << "scrap" << setw(10) << "cost" << endl;
    cout << " _____" << endl;
    for(int i=1;i<=years;i++)
    {
        cost-=scrap;
        cout << setw(10) << i << setw(10) << scrap << setw(10) << cost << endl;
    }
}
```

---