TIC1001—Introduction to Computing and Programming
National University of Singapore

Practice on Branching

Bigger Sum

Write a function `bigger_sum` that takes three numbers as arguments and returns the sum of the squares of the two larger numbers. For example, given three numbers 1, 2 and 3, since 2 and 3 are larger than 1, the `bigger_sum` function should return the integer value 13.

In [ ]:
```
int bigger_sum(int a, int b, int c)
{
    int sum, large;
    if (a >= b && c >= b)
    {
        sum = a*a + c*c;
    }
    else if (a >= c && b >= c)
    {
        sum = a*a + b*b;
    }
    else if (b >= a && c >= a)
    {
        sum = b*b + c*c;
    }
    return sum;
}
```

Leap Years

Write a function `is_leap_year` that takes one integer parameter and decides whether it corresponds to a leap year, i.e. `is_leap_year` returns true if the input parameter is a leap year, and false otherwise.

So which years are leap years? Well, accordingly to Wikipedia:

In the Gregorian calendar, the current standard calendar in most of the world, most years that are integer multiples of 4 are leap years. In each leap year, the month of February has 29 days instead of 28. Adding an extra day to the calendar every four years compensates for the fact that a period of 365 days is shorter than a solar year by almost 6 hours. This calendar was first used in 1582.

Some exceptions to this rule are required since the duration of a solar year is slightly less than 365.25 days. Over a period of four centuries, the accumulated error of adding a leap day every four years amounts to about three extra days. The Gregorian Calendar therefore omits 3 leap days every 400 years, omitting February 29 in the 3 century years (integer multiples of 100) that are not also integer multiples of 400. For example, 1600 was a leap year, but 1700, 1800 and 1900 were not. Similarly, 2000 was a leap year, but 2100, 2200, and 2300 will not be. By this rule, the average number of days per year is 365 + 1/4 − 1/100 + 1/400 = 365.2425.

In [ ]:
```
#include <stdbool.h>
bool is_leap_year(int year) {
    if (year % 4 ==0)
    {

        if (year % 400 == 0)
        {
            return 1;
        }
        else if (year % 100 == 0)
        {
            return 0;
        }
        else
        {
            return 1;
        }
       return 0;
    }
    else
    {
    return 0;
    }
}
```

I would like to buy a vowel

The five letters of the English alphabet A, E, I, O and U are called vowels. All other alphabets except these 5 vowel letters are called consonants.
Write a function is_vowel that takes a character as input and returns true is the character is a vowel and false otherwise. You may assume the input will always be an English letter.

In [ ]:
```
#include <stdbool.h>
bool is_vowel(char c)
{
    int vowel;

    if (c == a || c == A || c == e || c == E || c == i || c == I || c == o || c == O || c == u || c == U
    {
        vowel = 1;
    }
    else
    {
        vowel = 0;
    }
    return vowel;
}
```

Uber Fare

An Uber trip is calculated based on the following formula:

Distance covered	Cost
First 1 km	2.40
Every 200 m up to 10 km	0.10
Every 150 m after 10km	0.10

Write a function uber_fare that takes in the trip distance in metres and returns the cost of the trip in cents.

In [ ]:
```
int uber_fare(int distance) {
    double fare;
    if (distance <= 1000)
    {
        fare = 240;
    }
    else if (distance > 1000 && distance <= 10000)
    {
        if ((distance - 1000) / 200 < 1)
        {
            fare = 240 + 10;
        }
        else
        {
            fare = 240 + (((distance - 1000) / 200) * 10.0);
        }
    }
    else if (distance > 10000)
    {
        if ((distance - 10000) / 150 < 1)
        {
            fare = 240 + 500 + 10;
        }
        else
        {
            fare =  240 + 450 + (round((distance - 10000) / 150.0) * 10);
        }
    }
    return fare;
}
```

-END-