

Problem Statement

A1: being lapped on the track

You and your friend are running laps on the track at the rec complex. Your friend passes you exactly X times in one of your laps (that is, you start the lap together and at the X th time she passes you, it is at the end of your lap). How much faster in miles per hour is your friend going than you are? You only know the length of the track in miles per hour, the time in seconds for you to complete one lap, and X , the number of times she passes or comes next to you. Answer should include at least one decimal place if not an integer.

Example 1:

The track is 1 mile in length. You and your friend start together, she runs faster and passes you at the halfway point and again right at the end of the mile. It took you 900 seconds (15 minutes). You did 1 mile in .25 hours for 4 miles per hour. She ran 3 miles in 15 minutes for 12 miles per hour or 8 miles per hour faster than you. **Answer:** 8 miles/hour faster than you

Example 2:

Length = 0.5 mile

Time = 900 sec

$X = 1$

Answer: 2 miles per hour faster (she was 4 miles per hour, you were 2 miles per hour)

Problem Statement

2 Advanced — Probability

Write a program that takes as input the number n of games remaining in a certain baseball playoff series, the number k of wins needed for a specific team, and the probability p that this team will win a given game (this is a single probability that will remain the same for each game). The program will then output the probability that this team achieves at least the needed number of wins. To compute this probability, you can use the following formula (which assumes p is fixed):

$$f(n, k) =$$

8><>:

0 if $k > n$

1 if $k = 0$

$pf(n - 1, k - 1) + (1 - p)f(n - 1, k)$ otherwise

For example to compute the probability of winning at least 2 out of 3 games with a probability of 0.6 for a single game:

$$f(3, 2) = 0.6f(2, 1) + 0.4f(2, 2)$$

$$f(2, 1) = 0.6f(1, 0) + 0.4f(1, 1)$$

$$f(2, 2) = 0.6f(1, 1) + 0.4f(1, 2)$$

$$f(1, 0) = 1$$

$$f(1, 1) = 0.6f(0, 0) + 0.4f(0, 1)$$

$$f(1, 2) = 0$$

$$f(0, 0) = 1$$

$$f(0, 1) = 0$$

Then back-substituting:

$$f(1, 1) = 0.6$$

$$f(2, 2) = 0.6(0.6) = 0.36$$

$$f(2, 1) = 0.6 + 0.4(0.6) = 0.84$$

$$f(3, 2) = 0.6(0.84) + 0.4(0.36) = 0.648$$

You may assume that both n and k are nonnegative integers less than 10 and that $0 \leq p \leq 1$, (a probability of 0 means it is impossible, and a probability of 1 means it is guaranteed).

Example 1:

Enter number of games: 3

Enter number of wins: 2

Enter win probability: .6

Probability: 0.648

Example 2:

Enter number of games: 7

Enter number of wins: 4

Enter win probability: 0.45

Probability: 0.391712203125

Problem Statement

A3: Comparing currency exchanges

You are going to travel to France. You will need to convert dollars to euros (the currency of the European Union). There are two currency exchange booths. Each has a display that shows CR: their conversion rate as euros per dollar and their fee as a percentage. The fee is taken before your money is converted. Which booth will give you the most euros for your dollars, how many euros, and how much is the difference.

Example 1:

Dollars: 200

CR1: 0.78

Fee: 1 (amount 152.88 euros)

CR2: 0.80

Fee: 3 (amount 155.2 euros)

Answer: 2 is the best; difference is 2.32 euros; 155.2 euros

Example 2:

Dollars: 100

CR1: 1.15

Fee: 5 (amount 109.25 euros)

CR2: 1.10

Fee: 1 (amount 108.9 euros)

Answer: 1 is the best; difference is .3499 euros; 109.25 euros

Problem Statement

4 Advanced — Character Matching

Write a program that inputs two strings of the same length and displays an alignment of the strings that maximizes the number of positions in which the two strings contain the same character. For example, if the two strings are “computer” and “programs”, the optimal alignment is:

```
computer
```

```
programs
```

which has two matches: the ‘p’ in “computer” with the ‘p’ in “programs” and the ‘r’ in “computer” with the ‘r’ in “programs”. You may assume the two strings contain no blanks. If you are using a GUI to display your output, be sure to use a fixed-width font such as Courier New.

Example 1:

Enter string a: computer

Enter string b: programs

```
computer
```

```
programs
```

Example 2:

Enter string a: programs

Enter string b: computer

```
programs
```

```
computer
```

Example 3:

Enter string a: purple

Enter string b: people

```
purple
```

```
people
```

Problem Statement

A5: Average of last three tanks

Whenever the woman gets gas for her car, she fills the tank until the pump stops. She then records the current mileage and the amount of gas, e.g. 21007.9 miles and 14.5 gal. She likes to record the average miles per gal for the last three tanks of gas. Write a program that when given this data (that is, a list of at least 4 records of miles and gals - the first gas amount is ignored) will calculate the average miles per gal for tanks 2 to 4, tanks 3 to 5, ... and then the last three tanks. You may prompt for additional records. Answer should have at least one decimal place if not an integer.

Example 1: Odometer: 100.6 Gas: 30.4
 Odometer: 250.4 Gas: 30
 Answer: insufficient data

Example 2: Odometer: 1000 Gas: 20
 Odometer: 1250 Gas: 12
 Odometer: 1570 Gas: 10
 Odometer: 1950 Gas: 12 **Mileage: 27.9412**
 Odometer: 2260 Gas: 25 **Mileage: 21.4894**
 Odometer: 2700 Gas: 21 **Mileage: 19.4828**

Problem Statement

6 Advanced — Spiral

Consider a spiral path formed by walking through the Cartesian plane as follows:

- Begin at $(0, 0)$ facing toward $(1, 0)$.
- Repeat forever:
 - Take a step forward to the next grid point.
 - If the grid point to the right is not already in the path, turn right.

A portion of this spiral is shown in the following figure:

```
4
0
1
2
3
4
-1
-2
-3
-4
-4 -3 -2 -1 0 1 2 3
```

Write a program that takes as input a number of steps and outputs the grid point reached by this path after this number of steps. You may assume that the input is a nonnegative integer.

Example 1:

Enter number of steps: 0

Grid Point $(0, 0)$

Example 2:

Enter number of steps: 12

Grid Point $(2, -2)$

Example 3:

Enter number of steps: 1000000

Grid Point $(-500, -500)$

Solution
A1: Running on the track

```
int T;
float L;
float S1, S2, D;
int X;
cout << "\nEnter X: number of times lapped: ";
cin >> X;
cout << "\nEnter Time: ";
cin >> T;
cout << "\nEnter Length of track in miles ";
cin >> L;
S1 = 60*60*L/T;
S2 = 60*60*L*(X+1)/T;
D = S2 - S1;
cout << "\nSpeed difference is " << D;
cout << " S1 " << S1 << " S2 " << S2;
return 0;
```

```
/* 2 Advanced - Probability */
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Ksu.Contest2014.SeriesProbability
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter number of games: ");
            int games = Convert.ToInt32(Console.ReadLine());
            Console.Write("Enter number of wins: ");
            int wins = Convert.ToInt32(Console.ReadLine());
            Console.Write("Enter win probability: ");
            double winProb = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine();
            Console.WriteLine("Probability: " + Probability(wins, games, winProb));
            Console.ReadLine();
        }

        private static double Probability(int wins, int games, double winProb)
        {
            if (wins > games)
            {
                return 0;
            }
            else if (wins == 0)
            {
                return 1;
            }
            else
            {
                return winProb * Probability(wins - 1, games - 1, winProb) + (1 - winProb) * Probability
(wins, games - 1, winProb);
            }
        }
    }
}
```


Solution
A3: Money Changing

```
float D;
double E1, E2;
float CR1, CR2;
float F1, F2;
cout << "\nenter Dollars: ";
cin >> D;

cout << "\nenter booth 1 CR: conversion rate: ";
cin >> CR1;
cout << "\nenter booth 1 F: fee percentage: ";
cin >> F1;
E1 = (D - D*0.01*F1)*CR1 ;

cout << "\nenter booth 2 CR: conversion rate: ";
cin >> CR2;
cout << "\nenter booth 2 F: fee percentage: ";
cin >> F2;
E2 = (D - D*0.01*F2)*CR2 ;

if (E1 >= E2) { cout << "\nThe first exchange booth has the best
    amount. \nThe difference is " << E1 - E2; }
else {cout << "\nThe second exchange booth has the best amount.
    \nThe difference is " << E2 - E1; }

cout << "\nNumber of Euro from Booth One is " << E1;
cout << "\nNumber of Euro from Booth Two is " << E2;

return 0;
```

```
/* 4 Advanced - Character Matching */
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Ksu.Contest2014.MaxMatches
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter string a: ");
            string a = Console.ReadLine();
            Console.Write("Enter string b: ");
            string b = Console.ReadLine();
            string off1 = BestMatch(a, b);
            string off2 = BestMatch(b, a);
            if (Matches(a + off1, off1 + b) > Matches(off2 + a, b + off2))
            {
                Console.WriteLine(a + off1);
                Console.WriteLine(off1 + b);
            }
            else
            {
                Console.WriteLine(off2 + a);
                Console.WriteLine(b + off2);
            }
            Console.ReadLine();
        }

        private static int Matches(string a, string b)
        {
            int count = 0;
            for (int i = 0; i < a.Length; i++)
            {
                if (a[i] == b[i])
                {
                    count++;
                }
            }
            return count;
        }

        private static string BestMatch(string a, string b)
        {
            string padding = "";
            string bestMatch = "";
            int maxMatch = -1;
            for (int i = 0; i < a.Length; i++)
            {
                int matches = Matches(a + padding, padding + b);
                if (matches > maxMatch)
                {
                    bestMatch = padding;
                    maxMatch = matches;
                }
                padding += " ";
            }
            return bestMatch;
        }
    }
}
```

Solution
A5: Sliding Window Gas Average

```
float mile1, mile2,mile3,mile4;
float gas1, gas2, gas3, gas4;

char R = 'Y';

cout << "\nEnter first odometer reading: ";
cin >> mile1;
cout << "\nEnter first gas amount: ";
cin >> gas1;
cout << "\nEnter next odometer reading: ";
cin >> mile2;
cout << "\nEnter next gas amount: ";
cin >> gas2;
cout << "\nEnter next odometer reading: ";
cin >> mile3;
cout << "\nEnter next gas amount: ";
cin >> gas3;
cout << "\nEnter next odometer reading: ";
cin >> mile4;
cout << "\nEnter next gas amount: ";
cin >> gas4;
cout << "\nAverage over last 3 tanks is "<< (mile4 -
    mile1)/(gas2+gas3+gas4);
cout << "\nAre there more receipts? Y or N:";
cin >> R;

while (R != 'N'){
    gas1 = gas2; gas2 = gas3; gas3 = gas4;
    mile1 = mile2; mile2 = mile3; mile3 = mile4;

    cout << "\nEnter next odometer reading: ";
    cin >> mile4;
    cout << "\nEnter next gas amount: ";
    cin >> gas4;
    cout << "\nAverage over last 3 tanks is "<< (mile4 -
        mile1)/(gas2+gas3+gas4);
    cout << "\nAre there more receipts? Y or N:";
    cin >> R;
}

return 0;
```

```
/* 6 Advanced - Spiral */
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Ksu.Contest2014.SpiralAdvanced
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.Write("Enter number of steps: ");
            int n = Convert.ToInt32(Console.ReadLine());
            int i = 0;
            while (n > 0)
            {
                i++;
                n -= 8 * i;
            }
            Console.Write("Grid Point (");
            if (n < -6 * i)
            {
                Console.WriteLine(i + ", " + -(n + 7 * i) + ")");
            }
            else if (n < -4 * i)
            {
                Console.WriteLine(-(n + 5 * i) + ", " + -i + ")");
            }
            else if (n < -2 * i)
            {
                Console.WriteLine(-i + ", " + (n + 3 * i) + ")");
            }
            else
            {
                Console.WriteLine((n + i) + ", " + i + ")");
            }
            Console.ReadLine();
        }
    }
}
```