

UVA 10035 Aritmética Primaria

Práctica de Laboratorio N.02

UVA Judge

Submission Date: 9418603

Verdict: Accepted

Language: ANSI C

Run Time: 0.012

A los niños se les enseña a sumar números de varias cifras de derecha a izquierda, sumando una cifra cada vez. Muchos de ellos encuentran que la operación de “arrastre“, en la que se debe llevar un 1 de una posición a la siguiente, es todo un desafío. La tarea consiste en contar el número de operaciones de arrastre que se producirán en cada conjunto de problemas de suma, para ayudar a los educadores a evaluar la dificultad de los mismos.

Entrada

Cada línea de la entrada consta de dos enteros, sin signo, de menos de 10 cifras. La ultima línea de la entrada es “0 0“.

Salida

Por cada línea de la entrada, excepto la última, se debe calcular la cantidad de operaciones de arrastre que se producirán al sumar los dos números, dicha cantidad se mostrará en la salida.

1. Solución del problema

- 1.- Se declararon variables de tipo **unsigned int** debido a que solo manejaremos números no negativos.
- 2.- Se agregó el bucle while, con un scanf que llamará a dos valores y una condición de que ambos no sean ceros, lo que permitirá que los datos de entrada finalicen cuando no se cumpla esa condición.
- 3.- Se asignó el valor de 0 a dos variables: **e** y **sum**, los cuales en el bucle siguiente cambiarán sus valores, dependiendo de los datos de entrada que se hayan agregado.
- 4.- Este bucle hará que el valor de **e** sea sumado con cada par de dígitos de ambos datos de entrada desde la derecha hasta la izquierda.
- 5.- EL primer par de dígitos serán los restos correspondientes de dichos valores entre el valor de 10.
- 6.- Para que **e** vaya sumando cada par de dígitos desde la derecha hasta la izquierda, dentro del bucle los valores de entrada fueron eliminando su última cifra diviendose entre 10 y tomando el valor de su cociente (este proceso continuamente debido al bucle), mientras que las variables que iban tomando el valor del primer resto, fueron tomando el valor de los restos siguientes.
- 7.- Si la suma respectiva de **e** con cada par de dígitos es mayor o igual que 10, entonces ocurre un arrastre, el valor de **e** tomará luego de ello el valor de 0 cuando no ocurra un arrastre, y tomará 1 si es que ocurre un arrastre.
- 8.- El valor de **sum** indicará cuantas veces **e** toma el valor de 1, lo que quiere decir que nos indicará cuantas veces ocurre arrastre.
- 10.- El bucle tomó la condición de detenerse cuando ambos valores de entrada hayan eliminado todas sus cifras, convirtiéndose en 0.

- 11.- Al haber finalizado el primer bucle se colocó condiciones, en donde se imprimirá una sentencia dependiendo que valor toma la variable **sum**.
- 12.- EL bucle while principal seguirá pidiendo par de valores para operar y seguirá siendo así hasta que se rompa su condición, en pocas palabras, finalizará agregando **0 0**.

2. Código en C

```
1  #include <stdio.h>
2      int main(){
3          unsigned int a,b,sum,c,d,e;
4          while(scanf(" %u %u",&a,&b)>0&&((a!=0)||(b!=0))){
5              e=0;sum=0;
6              while((a!=0)||(b!=0)){
7                  c=a %10; d=b %10;
8                  if((c+d+e)>=10) {sum+=1; e=1;}
9                  else{e=0;}
10                 a=a/10;
11                 b=b/10;
12             }
13             if(sum==0) printf("No_carry_operation.\n");
14             else { if(sum==1) printf("1_carry_operation.\n");
15                   else{ printf(" %u_carry_operations.\n",sum);}
16             }
17         }
18         return 0;
19     }
```

Ejemplo Entrada

1571 5415
45218 454
57546468 864615
0 0

Ejemplo Salida

No carry operation.
1 carry operation.
5 carry operations.

UVA 10127 Unos

Práctica de Laboratorio N.02

UVA Judge

Submission Date: 9419148

Verdict: Accepted

Language: ANSI C

Run Time: 0.024

Para cualquier entero $0 \leq n \leq 10.000$ no divisible por 2 ni por 5, existe algún múltiplo del mismo que, en notación decimal, es una secuencia de unos. ¿Cuántos dígitos tiene el menor de tales múltiplos de n ?

3. Solución del problema

1.- Se declararon variables de tipo **long**, una tomará un valor por medio de un scanf, otros dos serán los que establezcan una condición y una operación, y el otro será el contador(el que indicará la cantidad de unos)

2.- Se agregó un bucle while el cual mantendrá pidiendo valores e imprimiendo su respectiva salida.(que sea diferente de EOF significa que parará hasta encontrar un caracter fin de archivo).

3.- Se asignó el valor 1 al contador, y a dos variables más.

4.- Se agregó el bucle **for** que mantendrá realizando la operación conocida de la división, en donde el número **n** es el divisor(dato de entrada), el Divisor principal será una secuencia de unos, pero para la determinación de cuantos unos posee tal divisor, el for mantendrá operando paso por paso, en donde se obtendrán varios residuos internos en la división y finalizará cuando el resto sea 0.

5.- El **for** resulta muy útil porque así no nos interesaría determinar el dividendo(que podría ser un número con una secuencia de unos muy larga), solo nos determinaría su cantidad de cifras, debido al contador.

6.- El contador tomará el valor de cuantos unos fueron necesarios del dividendo para mantener el for realizandose.

7.- Luego se imprimirá el contador, como dato de salida, para el respectivo datos de entrada.

4. Código en C

```
1  #include <stdio.h>
2  int main (){
3
4      long a,n,cont,b;
5      while(scanf("%ld", &n)!=EOF){
6          a=cont=b=1;
7          for(b=1;b=a%b){
8              if (a<n){
9                  a=(a*10)+1;
10                 cont++;
11             }
12             b=a%n;
13         }
14         printf("%ld\n",cont);
15     }
16
17     return 0;
18 }
```

Ejemplo Entrada

5461

9999

149

Ejemplo Salida

42

36

148

UVA 847 Un juego de multiplicación

Práctica de Laboratorio N.02

UVA Judge

Submission Date: 9439368

Verdict: Accepted

Language: ANSI C

Run Time: 0.004

En este problema se tiene que el juego empieza con 1 y los jugadores (sSan y Ollie). Empiezan multiplicando Stan con un numero del 2 al 9 y de ahí le toca a Ollie asta que uno pase el numero (N) que se define antes de empezar a jugar.

5. Solución del problema

1.- Se declaró una variable n de tipo **long double**, ya que el puede tomar un valor muy grande.

2.- De ahí se hizo un bucle for, en este se agregó un if el cual si i es impar se se le divide entre 9 y si es par se divide entre 2, y hasta que n sea menor que 1.

3.- Si al final i es par eso quiere decir q Ollie gana y si es impar Stan gana.

6. Código en C

```
1 #include <stdio.h>
2 int main(){
3
4     long double n;int i;
5     while(scanf("%Lf",&n)==1){
6         for(i=1;;i++){
7             if(i%2==1)n/=9;
8             else n/=2;
9             if(n<=1)break;
10        }
11        if(i%2==0)printf("Ollie_wins.\n");
12        else printf("Stan_wins.\n");
13    }
14    return 0;
15 }
```

Ejemplo Entrada

162

17

34012226

Ejemplo Salida

Stan wins.

Ollie wins.

Stan wins.

UVA 10082 WERTYU

Práctica de Laboratorio N.02

UVA Judge

Submission Date: 9439433

Verdict: Accepted

Language: ANSI C

Run Time: 0.004

En este problema se tiene q al momento de escribir el texto se colocan mal las manos por lo que se escribe una letra corrida a la derecha, por lo que el programa tiene que corregir el error.

7. Solución del problema

1.- Primero definimos la función **encuentra** y luego las cadenas de char(la funcion encuentra lee los caracteres).

2.- En la funcion principal main se hace un while en el cual se lee cada caracter de la cadena, y de ahí se hace el cambio de los caracteres.

8. Código en C

```
1  #include <stdio.h>
2  #include <string.h>
3  #define f(i,x,y) for(i = x; i<y; i++)
4  char a[] = "1234567890-= ";
5  char b[] = "QWERTYUIOP[]\\ ";
6  char c[] = "ASDFGHJKL;' ";
7  char d[] = "ZXCVBNM,./ ";
8  char cad[1000];
9  int i, j, m;
10 char res, x;
11
12 char encuentra (char fila [], char x){
13     f(j,1, strlen ( fila )){
14         if ( fila [j]==x) return fila[j-1];
15     }
16
17     return 0;
18 }
19
20 int main(){
21     while (1){
22         gets(cad);
23         m = strlen(cad);
24         if (m==0) break;
25         for(i = 0; i<m; i++){
26             x = cad[i];
27             res = encuentra(a, x);
28             if (res!=0) cad[i] = res;
29             res = encuentra(b, x);
30             if (res!=0) cad[i] = res;
31             res = encuentra(c, x);
32             if (res!=0) cad[i] = res;
33             res = encuentra(d, x);
34             if (res!=0) cad[i] = res;
35         }
36         puts(cad);
```

```
37     cad[0] = 0;
38     }
39 }
```

Ejemplo Entrada

O S, GOMR YPFSU/

Ejemplo Salida

I AM FINE TODAY.