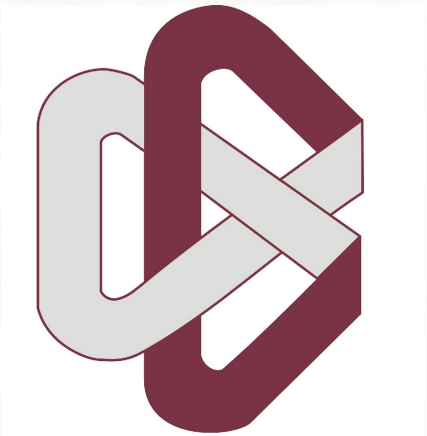


Informática Aplicada I

Marcela Morales Quispe

Octubre 23, 2013

Funciones y recursividad

Objevitos:

- Utilizar la función `rand()`.
- Comprender cómo escribir y utilizar funciones que se llamen a sí mismas.

Generación de números aleatorios

Considera el enunciado:

```
i=rand();
```

La función `rand` genera un entero entre **0** y **`RAND_MAX`,** si esta función en verdad produce enteros aleatorios, cualquier número entre **0** y **`RAND_MAX`** tiene la misma *oportunidad* (*o probabilidad*) de ser elegido, cada vez que `rand` es llamado.

El rango de valores producidos por rand, a menudo son distintos de los que se requieren en una aplicación específica. Por ejemplo: un programa que simule lanzar una moneda, lanzar dados, entre otros. ¿Cómo generar números aleatorios en el rango 1-6 para simular un juego de dados?

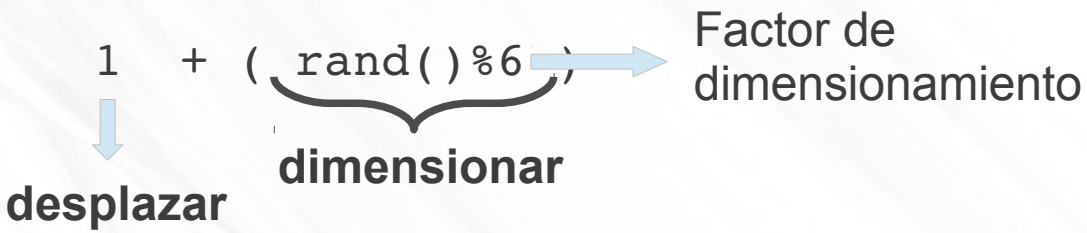
Dimensionar y desplazar enteros

Programa5-7.c

```
#include<stdio.h>
#include<stdlib.h>

int main()
{
      int i;
      for(i=1; i <=10; i++){
          printf("%10d", 1 + (rand()%6));

          if(i % 5 == 0)
               printf("\ln");
      }
      return 0;
}
```



¿Misma probabilidad?

Para mostrar que los números generados ocurren aproximadamente con la misma probabilidad, con el programa anterior simularemos 6000 tiradas de un dado. Cada entero del 1 al 6 debería de aparecer aproximadamente 1000 veces.

- Modifica el programa Programa5-7.c para esta simulación, guarda el nuevo programa bajo el nombre de Programa5-8.c

- **Tip**: puedes usar *contadores* para cada valor que se genera, es decir; si el número generado es 1, sumamos nuestro contador en 1 y así para cada valor generado, de tal forma que contemos todas las ocurrencias de cada valor.

- **Tip**: puede usar la sentencia **switch** en vez de muchos **if**'s anidados.

- Finalmente, muestra en forma de tabla las ocurrencias de cada valor.

¿En verdad son aleatorios?

Ejecuta nuevamente el programa: Programa5-7.c

- ¿La salida es la misma?
- ¿Cómo pueden ser aleatorios estos valores?

Irónicamente esta es una característica importante de la función **rand**, la cual sirve para depurar los programas.

Una vez que nuestros programas estén correctamente depurados podemos condicionar a que en cada ejecución se generen números aleatorios diferentes mediante el uso de la función **srand**, ésta toma como argumento un entero sin signo (**unsigned int**) para que en cada ejecución la función **rand** produca una secuencia diferente de números aleatorios. Se utiliza el especificador de conversión **%u** para leer un el valor con **scanf**.

1) Modifica el programa Programa5-7.c para pedir un entero sin signo, ejecuta y verifica que para cada ejecución se tienen salidas diferentes para diferentes semillas.
2) Modifica el programa Programa5-8.c y agrega la función **srand**, ¿La probabilidad de ocurrencia cambia?

Un juego de azar

Un jugador tira dos dados. Cada dado tiene 6 caras. Las caras contienen 1,2,3,4,5 y 6 puntos. Una vez que los dados se hayan detenido, se calcula la suma de los puntos en las dos caras superiores.

Si a la primera tirada, la suma es 7, o bien 11, el jugador gana. Si en la primera tirada la suma es 2, 3, 0 12 el jugador pierde (es decir, la casa "gana"). Si en la primera tirada, la suma es 4, 5, 6, 8, 9 o 10, entonces dicha suma se convierte en el "punto" o en la tirada. Para ganar, el jugador deberá continuar tirando los dados hasta que haga su "tirada". El jugador perderá si antes de hacer su tirada sale una tirada de 7.

Función para lanzar dos dados

```
#include<stdio.h> /*Completa la PRIMERA y SEGUNDA PARTE del programa*/
#include<stdlib.h>
#include<time.h>
int lanza2dados(void);
int main()
{
      int estado, sum, mipunt;
      srand(time(NULL));
      /*PRIMERA PARTE: lanzamos por primera vez y decidimos el estado del juego*/
      sum = lanza2dados();
      /*SEGUNDA PARTE: seguimos lanzando hasta ganar (o perder)*/
      /*TERCERA PARTE: Avisamos si ganamos o perdemos*/
      if(estado == 1)
        printf("El jugador gana!!!! =D\n");
      else
        printf("El jugador pierde .... =/\n");
      return 0;
}

int lanza2dados(void)
{
    int dado1, dado2, suma;
    dado1 = 1 + (rand() % 6);
    dado2 = 1 + (rand() % 6);
    suma = dado1 + dado2;
    printf("Dados %d + %d = %d\n", dado1, dado2, suma);
    return suma;
}
```

Funciones recursivas

Los programas que hemos analizado están estructurados en general como funciones que se llaman unas a otras, en forma disciplinada y jerárquica. Para algunos tipos de problemas, es útil tener funciones que se llamen a sí mismas.

Una ***función recursiva*** es una función que se llama así misma, ya sea directa, o indirecta a travéz de otra función.

Imagen tomada de: http://lizvillalobos.blogspot.mx/2010/04/recursividad.html

Ejercicios

- ***Factorial:*** de un número no negativo ***n*** (escrito como ***n!***), es el producto
  - *n*(n-1)*(n-2)*...*1*
- ***Serie Fibonacci:*** en su forma recursiva puede ser definidad como sigue:
  - fibonacci(0) = 0
  - fibonacci(1) = 1
  - fibonacci(n) = fibonacci(n-1) + fibonacci(n-2)

Bibliografía
Cómo programar en C/C++, H.M. Deitel, P.J. Deitel, 2da ed. Prentice Hall.