Random number generator

```c
#include <stdio.h>
#include <stdlib.h> /*header file for function rand(), srand();*/
int main () {
   int il;
   for( il = 1; il < 20; il = il + 1 ){
      printf("%10d", 1+rand()%6);
      if( il%5 == 0 ) printf("\n");
   }
   return 0;
}
```

There are functions defined in `stdlib.h` header file:

- `rand()` : This is a function which gives random numbers. (always less than `RAND_MAX` = 2147483647 in 64 bit machines)
- `srand(seed)` : This takes initial number as `unsigned int seed`.
- `srand(*seed)` : This function returns a random number in the range 0 to `RAND_MAX` just as rand does.

Rolling a six-sided die 10000 times

```c
#include <stdio.h>
#include <stdlib.h>
int main () {
   int i1, i2, i3, i4, i5, i6, ij, face;
   i1 = i2 = i3 = i4 = i5 = i6 = 0;
   for( ij = 1; ij <= 10000; ij = ij + 1 ) {
      face = 1 + rand() % 6;
      switch( face ) {
         case 1 :    ++i1;    break;
         case 2 :    ++i2;    break;
         case 3 :    ++i3;    break;
         case 4 :    ++i4;    break;
         case 5 :    ++i5;    break;
         case 6 :    ++i6;    break;
      }
   }
   printf("%s%13s\n", "Face", "Frequency");
   printf("   1%13d\n", i1);
   printf("   2%13d\n", i2);
   printf("   3%13d\n", i3);
   printf("   4%13d\n", i4);
   printf("   5%13d\n", i5);
   printf("   6%13d\n", i6);
   return 0;
}
```

Randomizing die-rolling program

```c
#include <stdio.h>
#include <stdlib.h>
int main () {
   int il;
   unsigned seed;
   printf("Enter seed: ");
   scanf("%u", &seed);
   srand(seed); /*now random number generation will start from number seed*/

   for( il = 1; il < 20; il = il + 1 ){
      printf("%10d", 1+rand()%6);
      if( il%5 == 0 ) printf("\n");
   }
   return 0;
}
```

A scoping example

```
#include <stdio.h>
void useLocal(void);
void useStaticLocal(void);
void useGlobal(void);
int x = 1; /* global variable */
int main () {
   int x = 5;
   printf("local x in outer scope of main is %d\n", x);
   {
      int x = 7;
      printf("local x in inner scope of main is %d\n", x);
   }
   printf("local x in outer scope of main is %d\n", x);
   useLocal();   useStaticLocal();   useGlobal();
   useLocal();   useStaticLocal();   useGlobal();
   printf("\nlocal x in main is %d\n", x);
   return 0;
}
void useLocal (void){
```

```c
void useLocal (void){
   int x = 25;
   printf("\nlocal x in useLocal is %d after entering useLocal\n", x);
   x++;
   printf("local x in useLocal is %d before exiting useLocal\n", x);
}
void useStaticLocal (void){
   static int x = 50;
   printf("\nlocal static x is %d on entering useStaticLocal\n", x);
   x++;
   printf("local static x is %d on exiting useStaticLocal\n", x);
}
void useGlobal (void){
   printf("\nglobal x is %d on entering useGlobal\n", x);
   x *= 10;
   printf("global x is %d on exiting useGlobal\n", x);
}
```