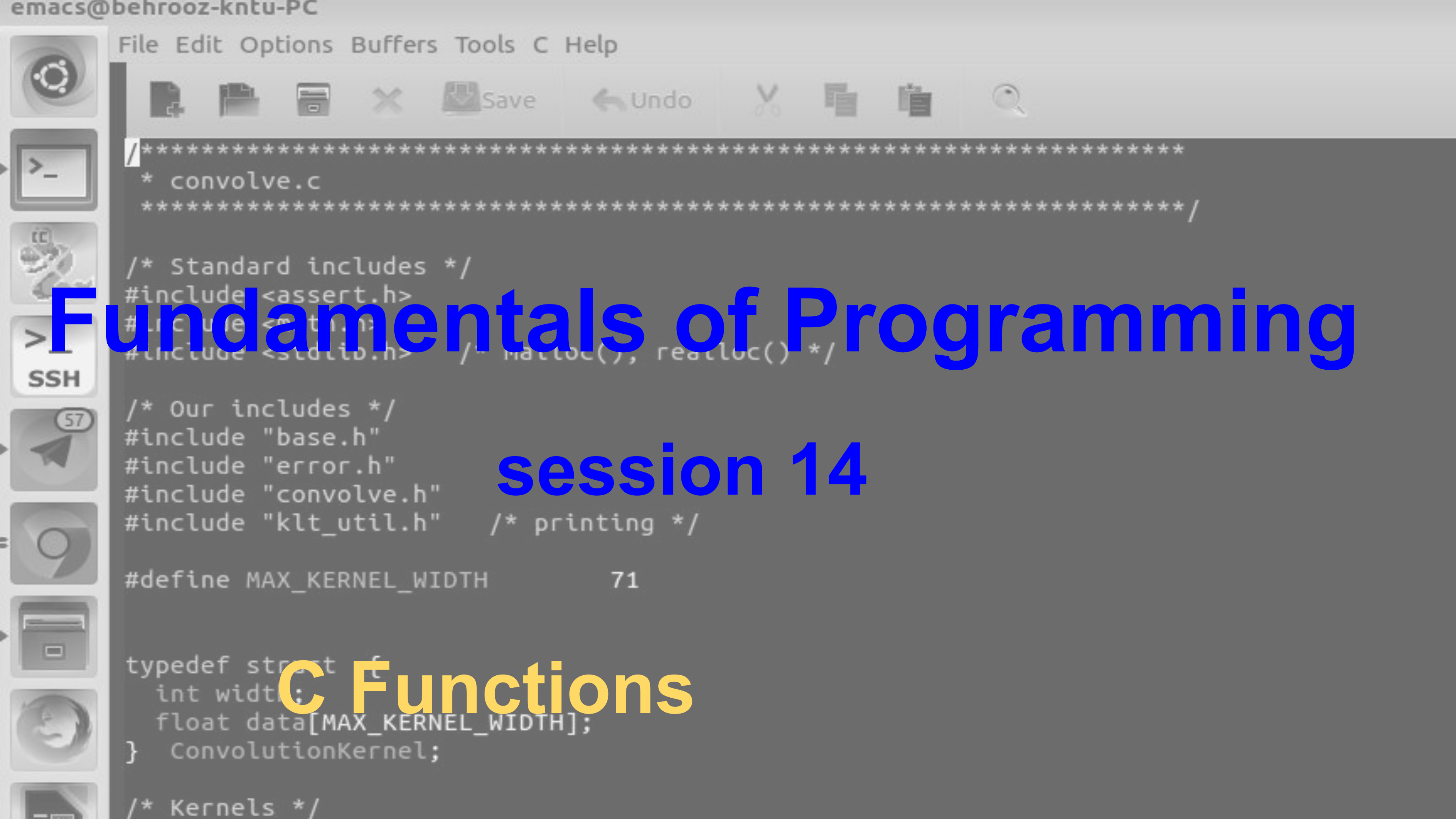
Fundamentals of Programming

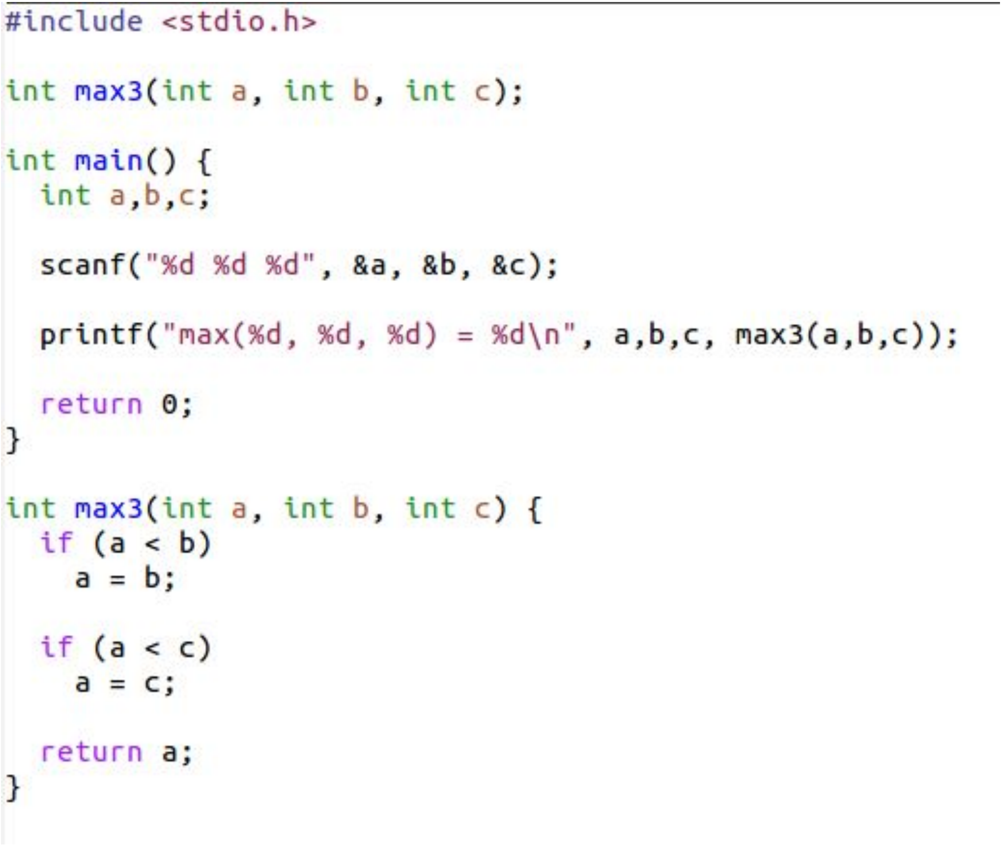
session 14

C Functions



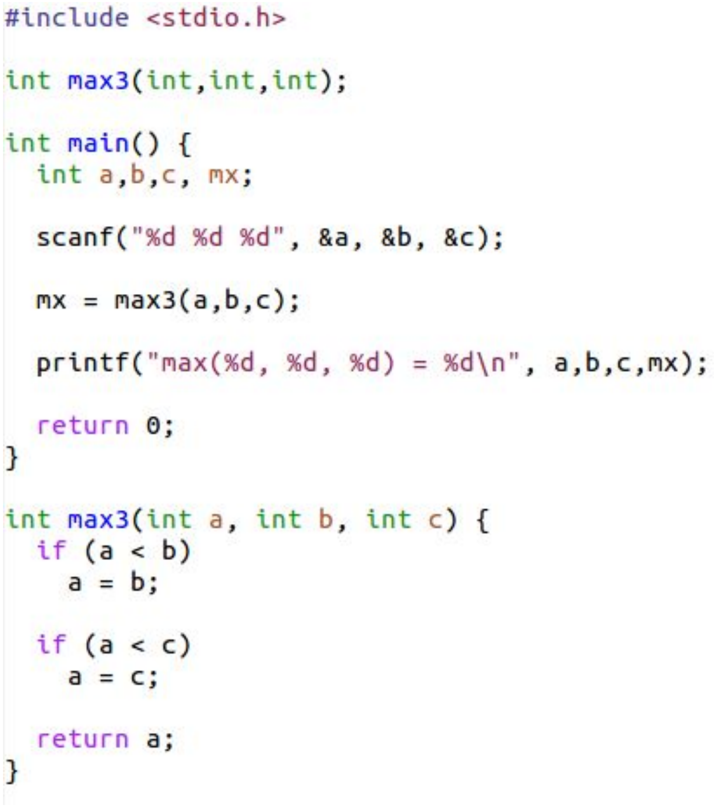
```
/**************************************************************
 * convolve.c
 **************************************************************/

/* Standard includes */
#include <assert.h>
#include <math.h>
#include <stdlib.h>    /* malloc(), realloc() */

/* Our includes */
#include "base.h"
#include "error.h"
#include "convolve.h"
#include "klt_util.h"   /* printing */

#define MAX_KERNEL_WIDTH        71


typedef struct {
  int width;
  float data[MAX_KERNEL_WIDTH];
}  ConvolutionKernel;

/* Kernels */
```

declarations / function prototypes

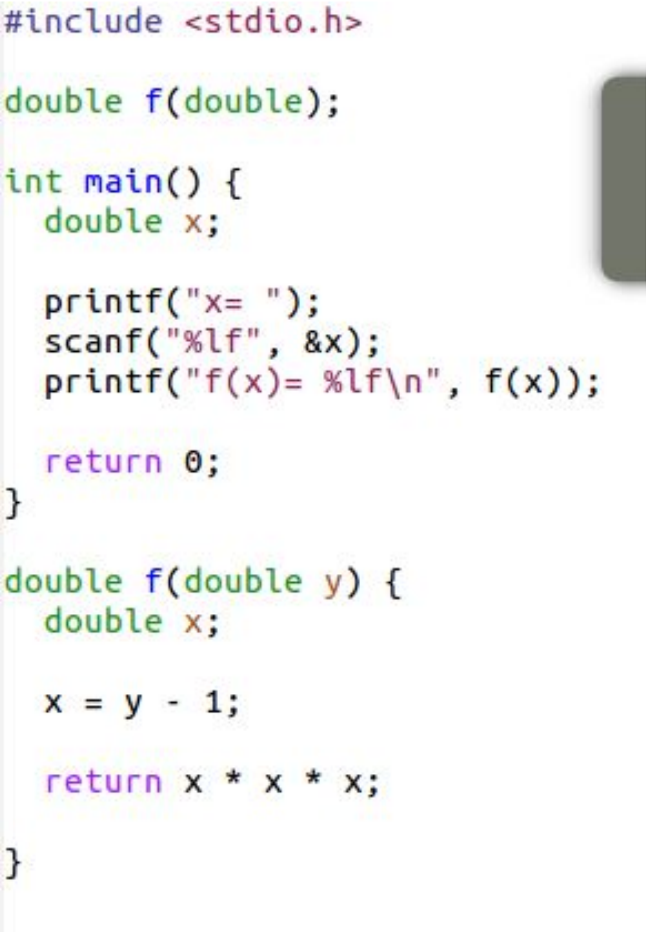


```c
#include <stdio.h>

int max3(int a, int b, int c);

int main() {
  int a,b,c;

  scanf("%d %d %d", &a, &b, &c);

  printf("max(%d, %d, %d) = %d\n", a,b,c, max3(a,b,c));

  return 0;
}

int max3(int a, int b, int c) {
  if (a < b)
    a = b;

  if (a < c)
    a = c;

  return a;
}
```

declarations / function prototypes



```c
#include <stdio.h>

int max3(int,int,int);

int main() {
  int a,b,c, mx;

  scanf("%d %d %d", &a, &b, &c);

  mx = max3(a,b,c);

  printf("max(%d, %d, %d) = %d\n", a,b,c,mx);

  return 0;
}

int max3(int a, int b, int c) {
  if (a < b)
    a = b;

  if (a < c)
    a = c;

  return a;
}
```



```
#include <stdio.h>

double f(double);

int main() {
  double x;

  printf("x= ");
  scanf("%lf", &x);
  printf("f(x)= %lf\n", f(x));

  return 0;
}

double f(double y) {
  double x;

  x = y - 1;

  return x * x * x;

}
```

C Preprocessor, include files

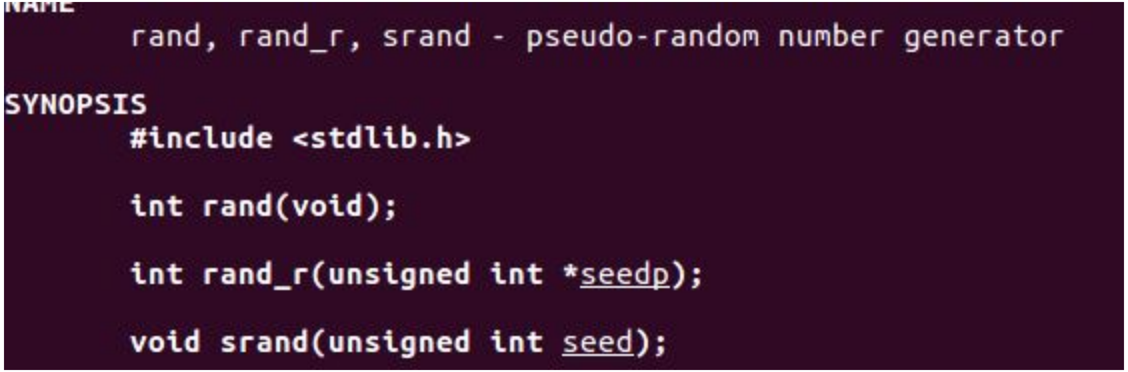
- printf (scanf, …)
  - library: **/lib/x86_64-linux-gnu/libc.so.6**
  - header file: **/usr/include/stdio.h**
    - **mostly contains function declarations**

C Preprocessor

- look at the header file
  - cat /usr/include/stdio.h
- look at the output of preprocessor:
  - gcc -E test.c

random number generation, rand, srand

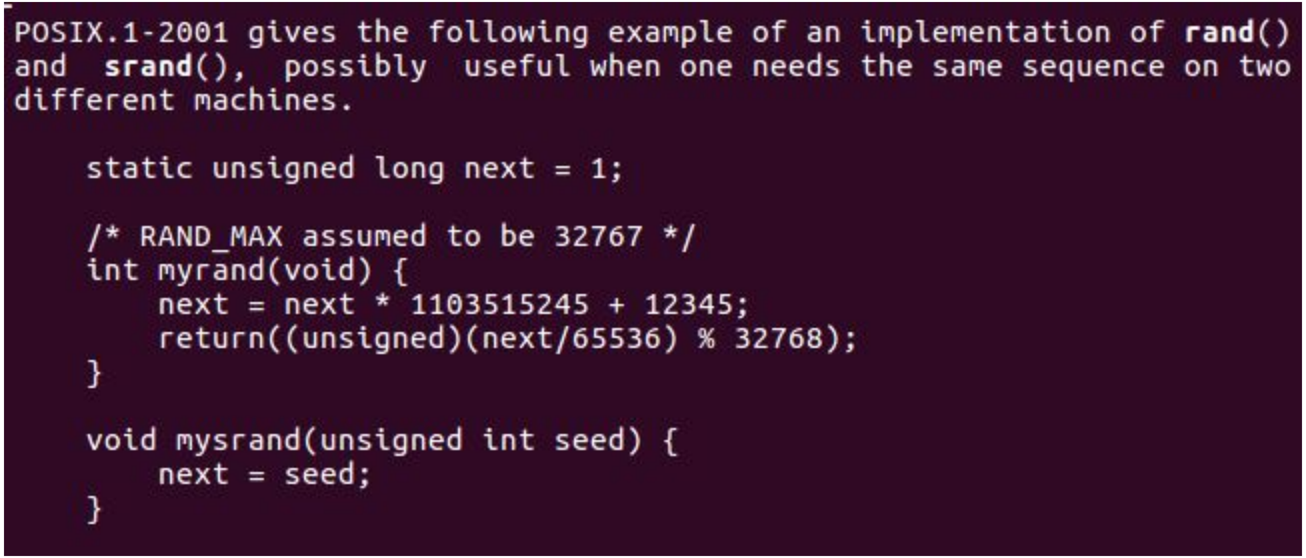
- run
  - man 3 rand



```
NAME
       rand, rand_r, srand - pseudo-random number generator

SYNOPSIS
       #include <stdlib.h>

       int rand(void);

       int rand_r(unsigned int *seedp);

       void srand(unsigned int seed);
```

random number generation, rand, srand

- also in **man 3 rand**



POSIX.1-2001 gives the following example of an implementation of **rand**() and **srand**(), possibly useful when one needs the same sequence on two different machines.

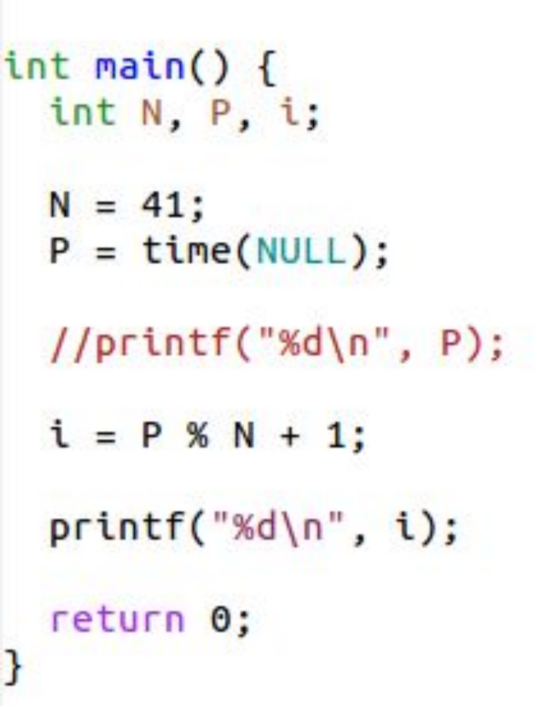
```
    static unsigned long next = 1;

    /* RAND_MAX assumed to be 32767 */
    int myrand(void) {
        next = next * 1103515245 + 12345;
        return((unsigned)(next/65536) % 32768);
    }

    void mysrand(unsigned int seed) {
        next = seed;
    }
```

random number generation, rand, srand

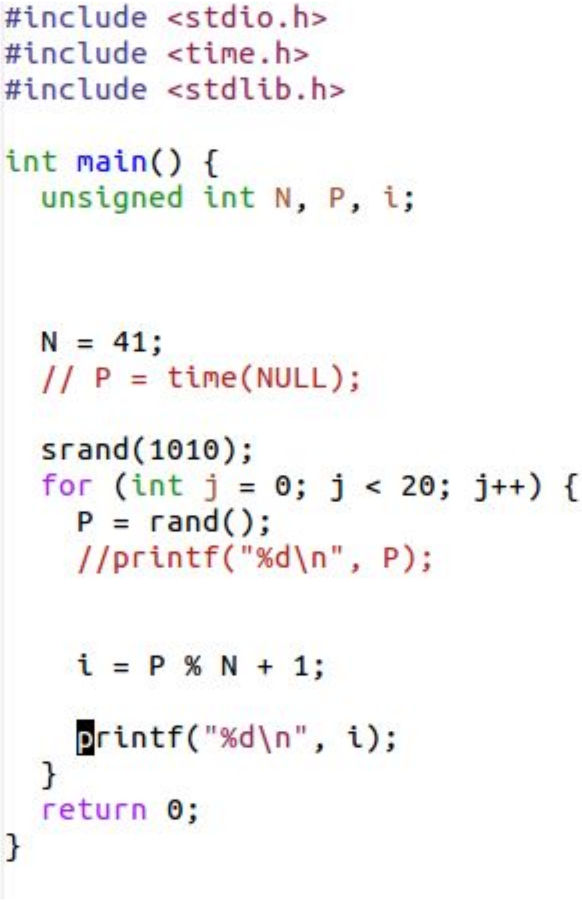
- a random number between **0** and **`RAND_MAX`**
  - `rand()`
- a random number between **0** and **n-1**
  - `rand() % n`
- a random number between **1** and **n**
  - `rand() % n + 1`

- **`RAND_MAX is defined in stdlib.h`**
  - `cat /usr/include/stdlib.h`
  - `#define  RAND_MAX    2147483647`

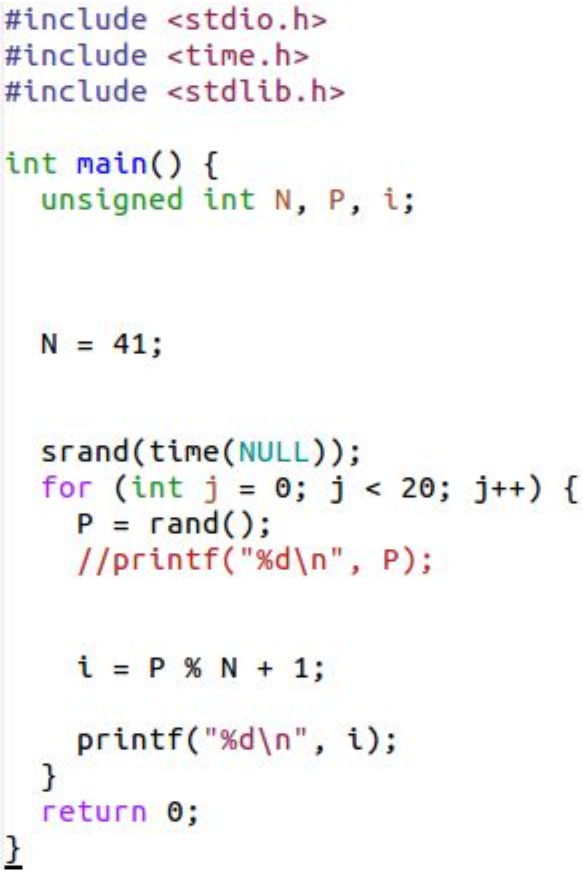
remember choose.c



```
int main() {
  int N, P, i;

  N = 41;
  P = time(NULL);

  //printf("%d\n", P);

  i = P % N + 1;

  printf("%d\n", i);

  return 0;
}
```



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

int main() {
  unsigned int N, P, i;



  N = 41;
  // P = time(NULL);

  srand(1010);
  for (int j = 0; j < 20; j++) {
    P = rand();
    //printf("%d\n", P);


    i = P % N + 1;

    printf("%d\n", i);
  }
  return 0;
}
```



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

int main() {
  unsigned int N, P, i;



  N = 41;


  srand(time(NULL));
  for (int j = 0; j < 20; j++) {
    P = rand();
    //printf("%d\n", P);


    i = P % N + 1;

    printf("%d\n", i);
  }
  return 0;
}
```