

**CSE1301  
Computer Programming:  
Lecture 20  
Case Study**

1

**Task**

- Design an algorithm and write a program that draws balls from a jar without replacement

2

**drawv1.c**

```
#include <stdio.h>
//draws numbers from a jar without replacement
int main ()
{
    // until the jar is empty or someone wins
    // pick a number and update the jar
    // call out the number
    return 0;
}
```

3

**drawv2.c**

<pre>#include &lt;stdio.h&gt; #define MAXNUM 75 int IsJarEmpty (int injar[]); int PickNum (int injar[]); void CallOut (int num); int main () {     int newnum;     int jar[MAXNUM];      while (!IsJarEmpty(jar))     {         newnum = PickNum(jar);         CallOut(newnum);     }     return 0; }</pre>	<pre>int IsJarEmpty (int injar[]) {     return 1; }  int PickNum (int injar[]) {     return 0; }  void CallOut (int num) {     return; }</pre>
---	--

4

**drawv3.c**

<pre>#include &lt;stdio.h&gt; #define MAXNUM 75 int IsJarEmpty (int injar[]); int PickNum (int injar[]); void CallOut (int num); int main () {     int newnum;     int jar[MAXNUM];      while (!IsJarEmpty(jar))     {         newnum = PickNum(jar);         CallOut(newnum);     }     return 0; }</pre>	<pre>int IsJarEmpty (int injar[]) {     return 0; //infinite loop }  int PickNum (int injar[]) {     return 0; }  void CallOut (int num) {     printf("Next number: %d\n",num);     return; }</pre>
---	---

5

**Algorithm for removing balls from a jar  
without replacement (PickNum)**

1. Pick a number between 1 and 75 at random
2. Remove the ball with this number from the jar array:
  - a. Push up in the array the remaining “balls”
  - b. Update the number of remaining balls

**Need another variable: numballs**

**Need to initialize the jar array!!**

**How does this affect IsJarEmpty?**

6

**drawv4.c**

```
#include <stdio.h>
#define MAXNUM 75
int InitializeJar(int injar[]);
int IsJarEmpty (int injar[]);
int PickNum (int injar[]);
void CallOut (int num);
int main ()
{
    int newnum, numballs;
    int jar[MAXNUM];
    numballs=InitializeJar(jar);
    while (!IsJarEmpty(jar))
    {
        newnum = PickNum(jar);
        CallOut(newnum);
    }
    return 0;
}

int InitializeJar(int injar[])
{
    ...
    return MAXNUM;
}

int IsJarEmpty (int injar[])
{
    return 1;
}

int PickNum (int injar[])
{
    return 0;
}

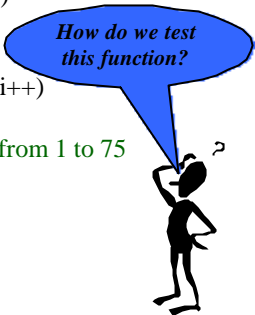
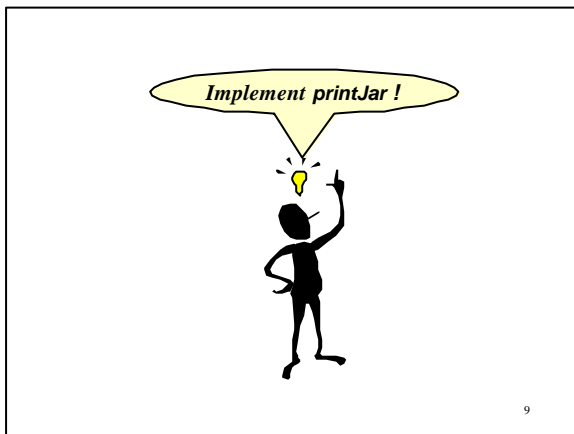
void CallOut (int num)
{
    printf("Next number: %d\n", num);
    return;
}
```

7

**drawv4.c -- InitializeJar**

```
int InitializeJar(int injar[])
{
    int i;
    for(i=0; i<MAXNUM; i++)
    {
        // bingo numbers run from 1 to 75
        injar[i]=i+1;
    }
    return MAXNUM;
}
```

8

**drawv5.c**

```
#include <stdio.h>
#define MAXNUM 75
int InitializeJar(int injar[]);
void PrintJar(int [], int);
int IsJarEmpty (int injar[]);
int PickNum (int injar[]);
void CallOut (int num);
int main ()
{
    int newnum, numballs;
    int jar[MAXNUM];
    numballs=InitializeJar(jar);
    printJar(jar,numballs);
    while (!IsJarEmpty(jar))
    {
        newnum = PickNum(jar);
        CallOut(newnum);
    }
    return 0;
}

int InitializeJar(int injar[])
{
    ...
    return MAXNUM;
}

void PrintJar(int injar[], int totalb)
{
    ...
    return;
}

int IsJarEmpty (int injar[])
{
    return 1;
}

int PickNum (int injar[])
{
    return 0;
}


void CallOut (int num)
{
    printf("Next number: %d\n", num);
    return;
}
```

10

**drawv5.c -- PrintJar**

```
void PrintJar(int injar[], int totalb)
{
    int i;
    for(i=0; i<totalb; i++)
    {
        printf("%d: %d\n", i, injar[i]);
    }
    return;
}
```


11



**How can we determine if the jar is empty?**

```
int IsJarEmpty (int totalb)
{
    return totalb == 0;
}
```

12



### drawv6.c

<pre>#include &lt;stdio.h&gt; #define MAXNUM 75 int InitializeJar(int injar[]); void PrintJar(int [], int); <b>int IsJarEmpty (int totalb);</b> int PickNum (int injar[]); void CallOut (int num); int main () {     int newnum, numballs;     int jar[MAXNUM];     numballs=InitializeJar(jar);     while(!<b>IsJarEmpty(numballs)</b>)     {         newnum = PickNum(jar);         CallOut(newnum);     }     return 0; }</pre>	<pre>int InitializeJar(int injar[]) {     ...     return MAXNUM; } void PrintJar(int injar[], int totalb) {     return; } <b>int IsJarEmpty (int totalb)</b> {     <b>return totalb==0;</b> } int PickNum (int injar[]) {     return 0; } void CallOut (int num) {     return; }</pre>
--	--

What will happen when we run this program?

13

Decrement numballs every iteration!

So how can we test IsJarEmpty?

14

### drawv6.c – Testing IsJarEmpty

<pre>#include &lt;stdio.h&gt; #define MAXNUM 75 int InitializeJar(int injar[]); void PrintJar(int [], int); <b>int IsJarEmpty (int totalb);</b> int PickNum (int injar[]); void CallOut (int num); int main () {     int newnum, numballs;     int jar[MAXNUM];     numballs=InitializeJar(jar);     while(!<b>IsJarEmpty(numballs)</b>)     {         newnum = PickNum(jar);         CallOut(newnum);         <b>numballs--;</b> //only for testing     }     return 0; }</pre>	<pre>int InitializeJar(int injar[]) {     ...     return MAXNUM; } void PrintJar(int injar[], int totalb) {     return; } <b>int IsJarEmpty (int totalb)</b> {     <b>return totalb==0;</b> } int PickNum (int injar[]) {     return 0; } void CallOut (int num) {     return; }</pre>
--	--

Can we test the program now?

15

### drawv7.c – Better Test for IsJarEmpty

<pre>#include &lt;stdio.h&gt; ... <b>int IsJarEmpty (int totalb);</b> int PickNum (int injar[]); void CallOut (int num, <b>int calln</b>); int main () {     int newnum, numballs;     int jar[MAXNUM];     <b>int callnum = 1;</b>     numballs=InitializeJar(jar);     while(!<b>IsJarEmpty(numballs)</b>)     {         newnum = PickNum(jar);         CallOut(newnum, <b>callnum</b>);         <b>numballs--;</b> //only for testing         <b>callnum++;</b>     }     return 0; }</pre>	<pre>int InitializeJar(int injar[]) {     ...     return MAXNUM; } void PrintJar(int injar[], int totalb) {     return; } <b>int IsJarEmpty (int totalb)</b> {     <b>return totalb==0;</b> } int PickNum (int injar[]) {     return 0; } void CallOut (int num, <b>int calln</b>) {     <b>printf("Call %d: %d\n", calln, num);</b>     return; }</pre>
--	--

16

### How do we pick a number and update the jar?

```
int PickNum (int injar[], int *totalb)
{
    int index, newnum;
    int i;
    // pick the index of the next number
    index = (*totalb) * rand()/(RAND_MAX+1);
    newnum = injar[index];
    // push up the remaining elements of the jar
    for(i=index; i<*totalb; i++)
    {
        injar[i] = injar[i+1];
    }
    (*totalb)--;
    return newnum;
}
```

Will this work?

17

No, because of integer division

18

```

drawv8.c -- PickNum
int PickNum (int injar[], int *totalb)
{
    int index, newnum;
    int i;
    double random;
    // pick the index of the next number
    random = rand();
    index = (*totalb) * random/(RAND_MAX+1.0);
    newnum = injar[index];
    // push up the remaining elements of the jar
    for(i=index; i< *totalb; i++)
    {
        injar[i] = injar[i+1];
    }
    (*totalb)--;
    return newnum;
}
    
```

```

drawv8.c -- PickNum
#include <stdio.h>
#include <stdlib.h>
...
int PickNum (int [], int *);
void CallOut (int num, int calln);
int main ()
{
    int newnum, numballs;
    int jar[MAXNUM];
    int callnum = 1;
     srand(1237);
    numballs=InitializeJar(jar);
    while(!IsJarEmpty(numballs))
    {
        newnum =
         PickNum(jar,&numballs);
        CallOut(newnum,callnum);
        numballs--; //only for testing
    }
    return 0;
}
    
```

*Careful!! We left the numballs-- instruction we used for testing*

```

drawv8.c -- Final Version
#include <stdio.h>
#include <stdlib.h>
...
int PickNum (int [], int *);
void CallOut (int num, int calln);
int main ()
{
    int newnum, numballs;
    int jar[MAXNUM];
    int callnum = 1;
    srand(1237);
    numballs=InitializeJar(jar);
    while(!IsJarEmpty(numballs))
    {
        newnum =
        PickNum(jar,&numballs);
        CallOut(newnum,callnum);
        callnum++;
    }
    return 0;
}
    
```

### A Different Approach

- Pre-assign the numbers randomly to the jar array
- Call out the numbers sequentially

```

drawv9.c -- Alternative Version
#include <stdio.h>
#include <stdlib.h>
#define MAXNUM 75
void InitializeJar (int injar[]);
int IsJarEmpty (int current);
int PickNum (int [], int);
void CallOut (int num, int calln);
int main ()
{
    int newnum, callnum=0;
    int jar[MAXNUM];
    srand(1237);
    InitializeJar(jar);
    while(!IsJarEmpty(callnum))
    {
        newnum =
        PickNum(jar, callnum);
        CallOut(newnum,callnum);
        callnum++;
    }
    return 0;
}
    
```

```

drawv9.c -- InitializeJar
void InitializeJar (int injar[])
{
    ...
    return;
}
    
```