

Handout 10

Arrays of integers

`int Values[number]` sets aside multiple spots in memory

- The names of the locations are `Values[0]`, `Values[1]`, ..., `Values[number - 1]`
- The locations are of type `int`
- The size of the array `number` must be an `int`

`double Values[number]` declares an array of doubles

10.1 First example of arrays

```
/*
First exploration of arrays
*/
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>

int main()
{
    int k=0; // counter
    int max=10; // size of array
    int values[max]; // array with int entries

    // fill the array
    while(k<max)
    {
        values[k] = k*k;
        k++;
    }
}
```

```

// print out the list of values in the array
k=0;
while (k<max)
{
    printf("Value number %d is %d\n", k, values[k]);
    k++;
}
}

```

10.2 Practice problem: Fibonacci numbers

Write a program that constructs the first 20 Fibonacci numbers. These numbers are defined by the condition that the first two Fibonacci numbers are both 1. Each subsequent number is the sum of the previous two numbers,

Your code should store the numbers in an array called `fib` and should print out the list. The result should be

```
1 1 2 3 5 8 13 21 34 55 89 144 233 ... 6765
```

10.3 Using arrays as counters

Fill in all of the question marks in this code, which you should call `flipcount.c`

```

/*
Using arrays to count coin flips
*/
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>

int main()
{
    // set up array to count heads & tails
    int flips[2];
    flips[0]=????;
    flips[1]=????;

    // setup for random coin flipping
    srand48(time(NULL));
    double coin;

    // flip coin 100 times, counting the number of heads & tails
    int k=1;

```

```

while(k<=100)
{
    coin = drand48();
    if (coin< ????)
        flips[0] = ????.;
    else
        flips[1] = ????.;
    k++;
}

// print results
printf("After 100 flips we find:\n");
printf("heads: ???????");
printf("tails: ???????");
}

```

10.4 Practice: dice-counter.c

Write a code `dice-counter.c` that uses random numbers to “roll a die” 100 times. Use an array to count how many times each of the numbers 1,2,3,4,5,6 is obtained. Finally, print out a table showing how many times each number was rolled.

10.5 Homework: find-mode.c

Write a piece of code that takes in 20 single-digit non-negative integers and reports which appears most in the following format:

```
The most frequent digit is ??, which appeared ?? times.
```

10.6 Challenge problem

One can think of running the code `flipcount.c` as a “game” where we flip 100 coins and take the difference between the number of heads and the number of tails.

Write a piece of code that “plays this game” 200 times. Print out a table:

heads - tails	count
>10	???
6 to 10	???
1 to 5	???
0	???
-5 to -1	???
-10 to -6	???
<-10	???