Computer Programming: Skills & Concepts (INF-1-CP1)

Loops

7th October, 2010

CP1–8 – slide 1 – 7th October, 2010
Summary of Lecture 7

- Summary of Practical 1
- The descartes.h package.
- Example program square.c
This Lecture

▶ The while statement.
▶ The for statement.
▶ fibonacci.c
▶ prime.c
▶ scanf and erroneous input.
while

while (<condition>) {
  <statement_sequence>;
}

while means “repeat until failure” (of the <condition>).
<statement-sequence> must alter some parameters involved in
<condition>. WHY?
Fibonacci Numbers

<table>
<thead>
<tr>
<th>n</th>
<th>a_n</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1+1 = 2</td>
</tr>
<tr>
<td>3</td>
<td>1+2 = 3</td>
</tr>
<tr>
<td>4</td>
<td>2+3 = 5</td>
</tr>
<tr>
<td>5</td>
<td>3+5 = 8</td>
</tr>
<tr>
<td>6</td>
<td>5+8 = 13</td>
</tr>
<tr>
<td>7</td>
<td>8+13 = 21</td>
</tr>
</tbody>
</table>
int main(void) {
    int n, next, count;
    int previous = 0;  /* Fibonacci -1 */
    int current = 1;   /* Fibonacci 0 */
    ...
    count = 0;

    while (count != n) {
        next = previous + current;  // i.e. 2 = 0 + 1
        previous = current;
        current = next;            // after: 2 + 1
        ++count;
    }
    printf("Fibonacci %d is %d", n, current);
    return EXIT_SUCCESS;
}
running fibonacci.c

: ./a.exe
Calculate which Fibonacci number? 0
Fibonacci 0 is 1

: ./a.exe
Calculate which Fibonacci number? 1
Fibonacci 1 is 1

: ./a.exe
Calculate which Fibonacci number? 2
Fibonacci 2 is 2

: ./a.exe
Calculate which Fibonacci number? 7
Fibonacci 7 is 21
while-statement: Repeat n-times

initialise iterator;
while (<not_iterator_endpoint>) {
<statement_sequence>;
next_iterator_value;
}

CP1–8 – slide 8 – 7th October, 2010
while-statement

Counting-up:

```c
count = 0;
while (count != n) {
    <statement_sequence>;
    ++count;
}
```

Counting-down:

```c
count = n;
while (count != 0) {
    <statement_sequence>;
    --count;
}
```
The for-loop

for (count = n; count != 0; --count) {
    <statement_sequence>;
}

CP1–8 – slide 10 – 7th October, 2010
Fibonacci using for

```c
int n, next, count;
int previous = 0; /* Fibonacci -1 */
int current = 1; /* Fibonacci 0*/
for (count = n; count != 0; --count) {
    next = previous + current;
    previous = current;
    current = next;
}
```
**Prime Numbers**

**Definition:** A prime number is any natural number which has no factors except itself and 1.

Prime: 3, 7, 11

Not Prime: 9 (3*3), 10 (2*5)

Simple test for primes:

\[ n \text{ is prime if } n=1 \text{ or if there is no integer } k \text{ between 2 and } \sqrt{n} \text{ such that } n \% k = 0. \]
prime.c

... 
k = 2; // First divisor-attempted is 2 
int prime = 1;
while (((k* k) <= n) && (prime)) { // finish at sqrt(n)
    if ((n % k) == 0) {
        printf("%d is %d * %d\n", n, n/k, k);
        prime = 0; // terminate the loop
    }
    ++k; // Test each value
}
if (prime)
    printf("%d is a prime number\n", n);
return EXIT_SUCCESS;