cpp-24

Loop Exercises in C

Three exercises for class practice: submit one or all of them for bonus points in Canvas. Use the pseudocode block to document your approach, and the "Lessons learnt" block below to record your observations or insights (if any).

Exercise 1: Sum of Even Numbers Using a `while` Loop

1. Write a C program that calculates the sum of even numbers between 1 and 100 inclusive using a while loop.

2. Pseudocode:

```
// Compute the sum of even numbers in [1,100] using `while`
Input: integers i in [1,100]
Output: sum of even numbers in [1,100]

Begin
// declare and init variables: sum, i
// while i <= 100:
  // if i even:
  // increase sum by i
  // increase i by 1
// print result
End
```

3. Code:

```
// declare variables `sum` (initialize) and `i` (loop)
int i=1, sum=0;
// while i <= 100
while (i <= 100) {
  // if i even, increase sum by i
  if (i % 2 == 0) sum+=i;
  // increase i by 1
  //printf("sum: %d, number: %d\n", sum, i++);
  i++;
 }
// print final result
printf("Sum of the even numbers in [1,100]: %d\n", sum);
```

```
Sum of the even numbers in [1,100]: 2550
```

As an arithmetic sequence:

```
// n: even numbers, a: starting number, l: last number
int n = 50, a = 2, l = 100;
printf("Sum of the even numbers in [1,100]: %d\n", n/2 * (a + l));
```

```
Sum of the even numbers in [1,100]: 2550
```

4. Lessons learnt:

    - It is necessary to print intermediate loop results.
    - The sum S of the first n terms of an arithmetic sequence is S = n/2 (a + l) where the first term is a = 2, the common difference d is 2, and the last term is l=100, or S = 50/2 (2 + 100) since n=(100-2)/2 + 1 = 50.

Exercise 2: Reverse a Number Using a `do-while` Loop

1. Write a C program that reverses the digits of a number using a do-while loop. For instance, if the input is 1234, the output should be 4321.

    Tip: 1234 = 10^0 * 4 + 10^1 * 3 + 10^2 * 2 + 10^3 * 1. It follows that 1234 % 10 = 4, 1234 / 10 = 123.4, 123 % 10 = 3, 123 / 10 =12.3 etc.

    ```
    printf("%d\n",1234%10);
    printf("%d\n",1234/10);
    printf("%d\n",123%10);
    printf("%d\n",123/10);
    printf("%d\n",12%10);
    printf("%d\n",12/10);
    printf("%d\n",1%10);
    printf("%d\n",1/10);
    ```

    ```
    4
    123
    3
    12
    2
    1
    1
    0
    ```

2. Pseudocode:

    ```
    // Reverse number digits
    Input: integer number
    Output: reversed number

    Begin
     // declare and initialize n=1234, nrev=0
     // do:
        // compute digit = n%10   (e.g. 1234 % 10 = 4, 123%10 = 3)
        // compute reverse nrev = nrev * 10 + digit (e.g. 0*10+4=4, 4*10+3= 43)
        // update n /= 10 (e.g. 1234/10 = 123)
        // while n > 0
     // print reversed number nrev
    End
    ```

3. Code

```
int n=1234, reverse = 0;
do {
  int digit = n % 10;
  reverse = reverse * 10 + digit;
  n /= 10;
 } while (n > 0);
printf("Reversed number: %d\n", reverse);
```

```
Reversed number: 4321
```

4. Lessons learnt:

   - It pays off to implement a short numeric example of the algorithm.
   - This algorithm makes use of the integer representation of floating-point numbers which processes 1.23 as 1.

```
printf("Uses %d but real value is %g\n",1234/10, 1234./10.);
```

```
Uses 123 but real value is 123.4
```

Exercise 3: Calculate the Factorial of a Number Using a `for` Loop

Write a C program that calculates the factorial of a number using a for loop. The factorial of a number n is the product of all positive integers less than or equal to n, e.g. for n = 4: 4! = 4 * 3 * 2 * 1. 0! is defined to be 1.

1. Pseudocode:

```
// Compute the factorial n! of an integer n > 0
Input: integer n
Output: factorial n! = n * (n-1) * ... * 1

Begin:
   // declare and initialize n and factorial fac

   // loop from i = 1 to i = n
     // compute factorial fac = fac * i
   // print result
```

2. Code:

```
// declare and initialize n and factorial fac
int n = 4, fac = 1;
// loop from i = 1 to i = n
for (int i = 1; i <= n; i++) {
  // compute factorial fac
  fac *= i;
  // print result
  printf("i = %d, factorial = %d\n", i, fac);
  }
printf("The factorial of %d is %d\n", n, fac);
```

```
i = 1, factorial = 1
i = 2, factorial = 2
i = 3, factorial = 6
i = 4, factorial = 24
The factorial of 4 is 24
```

1. Lessons learnt:
   - print loop variables first to make sure

Author: marcus

Created: 2024-04-12 Fri 08:03

Validate