Chapter 2: Pseudocode

WMC CS CLUB

Nov 25, 2021

§1 Introduction to Programs

Frequently, one must use or modify sections of another programmer's code. Since the original author is often unavailable to explain his/her code, and documentation is, unfortunately, not always available or sufficient, it is essential to be able to read and understand an arbitrary program.
The programs are written using a pseudocode that should be readily understandable by all programmers familiar with a high-level programming language, such as Python, Java, or C.

§2 Description of the ACSL Pseudo-code

We will use the following constructs in writing this code for this topic in ACSL:

Operators
! (not) , $\wedge$ or $\uparrow$(exponent), *, / (real division), % (modulus), +, −, >, <, >=, <=, !=, ==, && (and), || (or) in that order of precedence.

Functions
`abs(x)` - absolute value, `sqrt(x)` - square root, `int(x)` - greatest integer <= x.

Variables
Start with a letter, only letters and digits.

Sequential statements
```
INPUT variable
variable = expression (assignment)
OUTPUT variabl
```

Decision statements
```
IF boolean expression THEN
Statement(s)
ELSE (optional)
Statement(s)
END IF
```

Indefinite Loop statements
`WHILE Boolean expression Statement(s) END WHILE`

Definite Loop statements
```
FOR variable = start TO end STEP increment
Statement(s)
NEXT
```

Arrays
1 dimensional arrays use a single subscript such as $A(5)$. 2 dimensional arrays use (row, col) order such as $A(2,3)$. Arrays can start at location 0 for 1 dimensional arrays and location $(0,0)$ for 2 dimensional arrays. Most ACSL past problems start with either $A(1)$ or $A(1,1)$. The size of the array will usually be specified in the problem statement.

Strings
Strings can contain 0 or more characters and the indexed position starts with 0 at the first character. An empty string has a length of 0. Errors occur if accessing a character that is in a negative position or equal to the length of the string or larger. The len(A) function finds the length of the string which is the total number of characters. Strings are identified with surrounding double quotes. Use [ ] for identifying the characters in a substring of a given string as follows:
S = "ACSL WDTPD" (S has a length of 10 and D is at location 9)
S[:3] = "ACS" (take the first 3 characters starting on the left)
S[4:] = "DTPD" (take the last 4 characters starting on the right)
S[2:6] = "SL WD" (take the characters starting at location 2 and ending at location 6) S[0] = "A" (position 0 only).
String concatenation is accomplished using the + symbol

§3 Practice Problems

§3.1 Problems

1. After this program is executed, what is the value of B that is printed if the input values are 50 and 10?

```
input H, R
B = 0
if H>48 then
    B = B + (H - 48) * 2 * R
    H = 48
end if
if H>40 then
   B = B + (H - 40) * (3/2) * R
   H = 40
end if
B = B + H * R
output B
```

2. After the following program is executed, what is the final value of NUM?

```
A =     BANANAS
NUM = 0: T =
for J = len(A) - 1 to 0 step   1
     T = T + A[j]
next
for J = 0 to len(A) - 1
    if A[J] == T[J] then NUM = NUM + 1
next

```

3. After the following program is executed, what is the final value of C[4]?

```
A(0) = 12: A(1) = 41: A(2) = 52
A(3) = 57: A(4) = 77: A(5) = -100
B(0) = 17: B(1) = 34: B(20 = 81
J = 0: K = 0: N = 0
while A(J) > 0
  while B(K) <= A(J)
    C(N) = B(K)
    N = N + 1
    k = k + 1
  end while
  C(N) = A(J): N = N + 1: J = J + 1
end while
C(N) = B(K)

```

§3.2 Answer Key

1. The final value of B is $2 * 2 * 10 + 8 * 3/2 * 10 + 40 * 10 = 40 + 120 + 400 =$ **560**.

2. The program first stores the reverse of variable A into variable T and then counts the number of letters that are in the same position in both strings. Variable NUM is incremented each time a character at position x of A is the same as the character in position x of string T. There are 5 such positions: 1, 2, 3, 4, and 5.

3. The value of C[4] is **52**.