Solution Manual for Invitation to Computer Science 6th Edition Schneider 1133190820 9781133190820

Solution – Pseudocode

Part 1 – Writing Pseudocode

1.   a. 50 x 5 x $5 + $0 for gravel + 0 x $100 for parking = $1,250

   b. 100 x 10 x $5 + $500 for asphalt + 2 x $100 for parking = $5,700

   c. 600 x 10 x $5 + $1,000 for concrete + 8 x $100 for parking = $31,800

   a. 10 x 5 x $5 + $500 for asphalt + 0 x $100 for parking = $750, so $1,000

2. Get a value for `length` in feet
   Get a value for `width` in feet
   Get a value for `material`
   Get a value for `parking`

   Set `cost` to `length * width * 5`
   If (`material` = "asphalt") Then
     Set `cost` to `cost + 500`
   If (`material` = "concrete") Then
     Set `cost` to `cost + 1000`
   Set `cost` to `cost + (parking * 100)`
   If (`cost` < 1000) Then
     Set `cost` to `1000`

   Print "Total cost is"
   Print `cost`

Part 2 – Reading Pseudocode

1.

   a. Your change is
      0 quarters

1 dimes
0 nickels
4 pennies

b. Your change is
1 quarters
0 dimes
0 nickels
0 pennies

c. Your change is
3 quarters

2 dimes
1 nickels
4 pennies

2. This algorithm will determine how to make change for an amount of money in US currency. In this case, it will start with all pennies, then attempt to substitute larger coins up through quarters.

3. Depending on how comfortable students are with modulus and integer division, there are a couple different ways to approach making the algorithm better. The first answer is longer, but it doesn't use modulus and integer division.

Get a value for `change` (in cents)
Set `pennies` to `change`
Set `nickels, dimes, quarters` to 0

While (`change` >= 25)
  Set `quarters` to `quarters` + 1
  Set `change` to `change` - 25

While (`change` >= 10)
  Set `dimes` to `dimes` + 1 Set
  `change` to `change` - 10

While (`change` >= 5)
  Set `nickels` to `nickels` +
  1 Set `change` to `change` - 5

Set `pennies` to `change`

Print "Your change is"
Print the value of `quarters` and "quarters"
Print the value of `dimes` and "dimes" Print
the value of `nickels` and "nickels" Print
the value of `pennies` and "pennies"

If you are comfortable with modulus and integer division, you could

do Get a value for `change` (in cents)

Set `quarters` to `change / 25`
Set `change` to `change mod 25`
Set `dimes` to `change / 10`
Set `change` to `change mod 10`
Set `nickels` to `change / 5`
Set `pennies` to `change mod 5`

Print "Your change is"
Print the value of `quarters` and "quarters"
Print the value of `dimes` and "dimes"
Print the value of `nickels` and "nickels"
Print the value of `pennies` and "pennies"

4. Get a value for `change` (in cents)

Get a value for `quarters` in machine
Get a value for `dimes` in machine Get
a value for `nickels` in machine Get
a value for `pennies` in machine

While (`change` >= 25 and `quarters` >
   0) Set `quarters` to `quarters` - 1
   Set `change` to `change` - 25

While (`change` >= 10 and `dimes` > 0)
   Set `dimes` to `dimes` - 1
   Set `change` to `change` - 10

While (`change` >= 5 and `nickels` > 0)
   Set `nickels` to `nickels` - 1
   Set `change` to `change` - 5

If (`pennies` >= `change`) Then
   Print "There is enough money to make change"
Else
   Print "There is not enough money to make change"

Solution – Flowcharts

1.

2. You would need to keep track of how many times each player has won. After a victory you would need to update the wins for the winning player for the round and check to see if that player now has 3 wins. If not, loop back and play another round, much like if there was a tie previously.

Start

Set P1 & P2 wins to 0

Ask P1 for move

Ask P1 for move

Did P1 choose rock? — No → Did P1 choose paper? — No

Yes

Did P2 choose rock? — No → Did P2 choose paper? — No
Yes — Yes

Did P2 choose rock? — No → Did P2 choose paper? — No
Yes — Yes

Did P2 choose rock? — No → Did P2 choose paper? — No
Yes — Yes

Add 1 to P1 wins

Add 1 to P2 wins

It is a tie

Does P1 have 3 wins? — No

Does P2 have 3 wins? — No

Yes

Yes

P1 is the winner → End ← P2 is the winner