

ITI1120 Section 7 Exercise Solutions

Program Memory

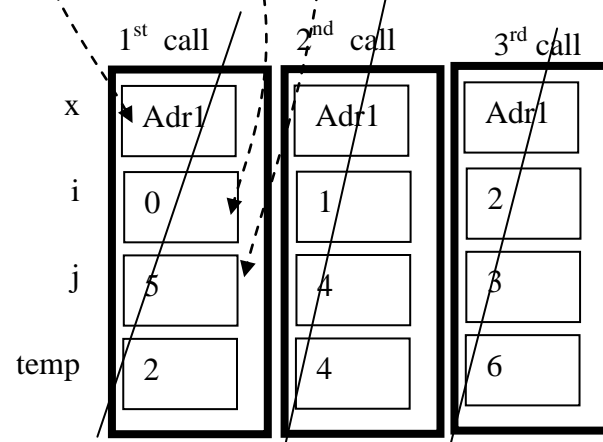
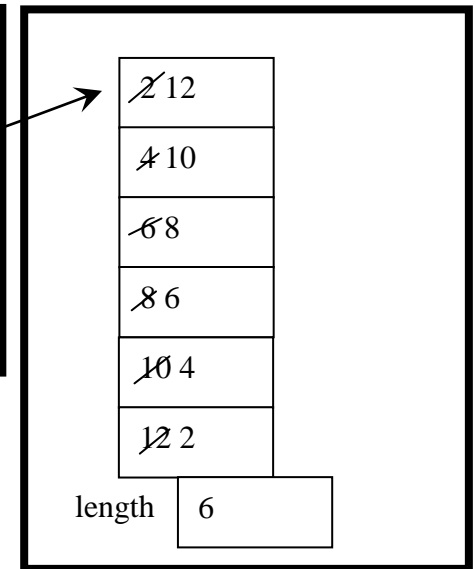
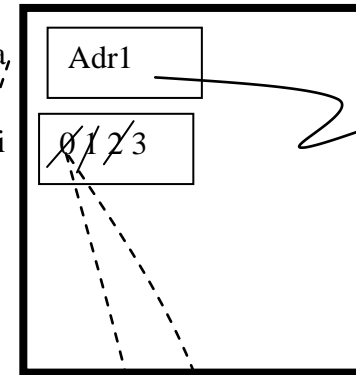
Exercise 7-1- Swap Values in an Array

Working memory

Global Memory

```

class SwapTilYouDrop
{
    public static void main (String args[ ])
    { int i = 0;
      int a;
      a = new int [
                { 2, 4, 6, 8, 10, 12 } ;
      while( i <= 2 )
      {
          arraySwap(a, i, 5 - i ) ;
          i = i + 1;
      }
      for ( i = 0 ; i <= 5 ; i = i + 1 )
      { System.out.println("a[" + i + "] is "
        + a[i] );
      }
    }
    // arraySwap : swaps values at i,j
    // Givens: x, an array,
    //         i,j, 2 indices in x
    public static void arraySwap(int[ ] x,
                                  int i,int j)
    {
        // DECLARE VARIABLES
        int temp ; // Inter: holds x[i]
        // BODY OF ALGORITHM
        temp = x[i] ;
        x[i] = x[j] ;
        x[j] = temp;
    }
}
    
```



Terminal Window

```


a[0] is 12
a[1] is 10
a[2] is 8
a[3] is 6
a[4] is 4
a[5] is 2
    
```

CPU

Exercise 7-1: Trace – Table 1, page 1, main

Statement	i	a	Array	Output
Initial values	?	?	?	
1. <code>i = 0</code>	0		?	
2. <code>a = new int[] {2,4,6,8,10,12}</code>			{2,4,6,8,10,12}	
3. <code>while (i <= 2) true</code>				
4. <code>arraySwap(a, i, 5-i)</code> see Table 2			{12,4,6,8,10,2}	
5. <code>i = i + 1</code>	1			
3. <code>while (i <= 2) true</code>				
4. <code>arraySwap(a, i, 5-i)</code> see Table 3			{12,10,6,8,4,2}	
5. <code>i = i + 1</code>	2			
3. <code>while (i <= 2) true</code>				
4. <code>arraySwap(a, i, 5-i)</code> see Table 4			{12,10,8,6,4,2}	
5. <code>i = i + 1</code>				
3. <code>while (i <= 2) false</code>	3			

Exercise 7-1: Trace – Table 1, page 2, main

Statement	i	a	Array	Output
(most recent values from page 1)	3		{12,10,8,6,4,2}	
6. for (i = 0; i <= 5; i = i + 1)	0			
7. System.out.println("A["+i+"] = "+a[i])				a[0] = 12
6. for (i = 0; i <= 5; i = i + 1) true	1			
7. System.out.println("A["+i+"] = "+a[i])				a[1] = 10
6. for (i = 0; i <= 5; i = i + 1) true	2			
7. System.out.println("A["+i+"] = "+a[i])				a[2] = 8
6. for (i = 0; i <= 5; i = i + 1) true	3			
7. System.out.println("A["+i+"] = "+a[i])				a[3] = 6
6. for (i = 0; i <= 5; i = i + 1) true	4			
7. System.out.println("A["+i+"] = "+a[i])				a[4] = 4
6. for (i = 0; i <= 5; i = i + 1) true	5			
7. System.out.println("A["+i+"] = "+a[i])				a[5] = 2
6. for (i = 0; i <= 5; i = i + 1) false	6			

Exercise 7-1: Trace – Table 2, arraySwap(a, 0, 5)

arraySwap(a, i, 5-i)

Adr1 0 5

↓ ↓ ↓

arraySwap(x, i, j)

Statement	x	i	j	temp	Array in Table 1
Initial values	2	0	5		{2,4,6,8,10,12}
1. temp = x[i]				2	
2. x[i] = x[j]					{12,4,6,8,10,12}
3. x[j] = temp					{12,4,6,8,10,2}

Exercise 7-1: Trace – Table 3, arraySwap(a, 1, 4)

arraySwap(a, i, 5-i)

Adr1 1 4

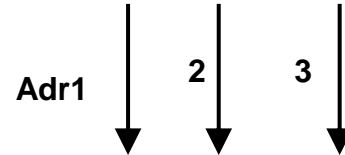
↓ ↓ ↓

arraySwap(x, i, j)

Statement	x	i	j	temp	Array in Table 1
Initial values	12	1	4		{12,4,6,8,10,2}
1. temp = x[i]				4	
2. x[i] = x[j]					{12,10,6,8,10,2}
3. x[j] = temp					{12,10,6,8,4,2}

Exercise 7-1: Trace – Table 4, arraySwap(a, 2, 3)

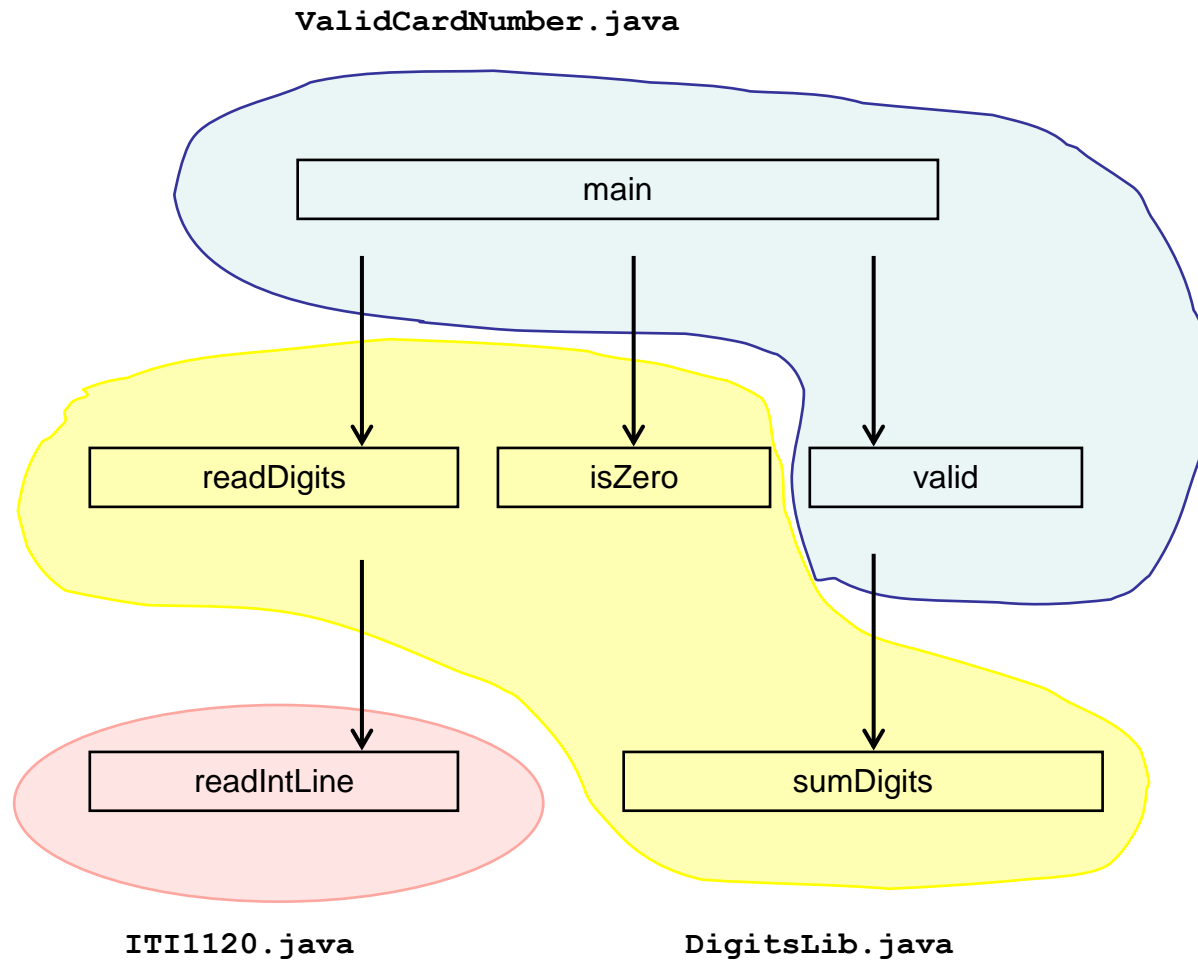
arraySwap(a, i, 5-i)

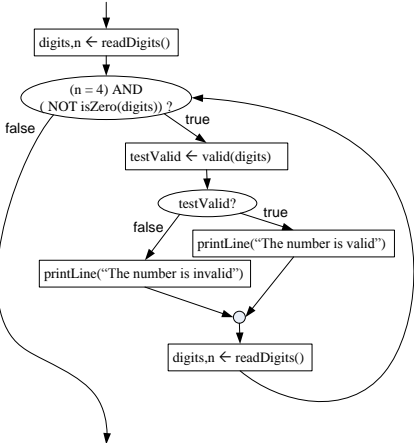


arraySwap(x, i, j)

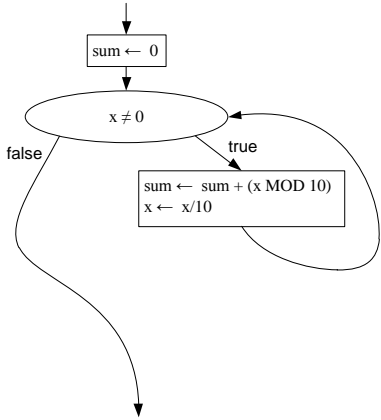
Statement	x	i	j	temp	Array in Table 1
Initial values		2	3		{12,10,6,8,4,2}
1. temp = x[i]				6	
2. x[i] = x[j]					{12,10,8,8,4,2}
3. x[j] = temp					{12,10,8,6,4,2}

Exercise 7-2 : Validating Numbers



Algorithm Model	Java
ValidCardNumber Class (ValidCardNumber.java)	
<p>GIVENS: (none) RESULT: (none) INTERMEDIATE: digits (reference to an integer array of digits) n (number of elements in the array) testValid (Indicated if card is valid) HEADER: main BODY:</p>  <pre> graph TD Start(()) --> Read1[digits, n ← readDigits()] Read1 --> Cond1{ (n = 4) AND (NOT isZero(digits))? } Cond1 -- true --> Valid[testValid ← valid(digits)] Cond1 -- false --> PrintInvalid[printLine("The number is invalid")] Valid --> Cond2{ testValid? } Cond2 -- true --> PrintValid[printLine("The number is valid")] Cond2 -- false --> PrintInvalid PrintValid --> Merge(()) PrintInvalid --> Merge Merge --> Read2[digits, n ← readDigits()] Read2 --> Cond1 </pre>	<pre> /* the main method calls the other methods in order to solve particular tasks. */ public static void main (String [] args) { int [] digits; // reference variable to array of digits boolean testValid; // indicates if card is valid // body digits = DigitsLib.readDigits(); // call to read the data while ((digits.length == 4) && (!DigitsLib.isZero(digits))) { // sends the number to the valid() method testValid = valid(digits); // print the result if (testValid) { System.out.println("This number is valid."); } else { System.out.println("This number is invalid."); } digits = DigitsLib.readDigits(); } } </pre>
<p>GIVENS: digits (reference to array of digits) n (number of elements in the array) RESULT: isValid (TRUE if card number is valid, FALSE otherwise) INTERMEDIATE: firstThree (first three digits of last group) lastDigit (last digit of credit card number) sum (sum of 15 digits) HEADER: isValid ← valid(digits, n) BODY:</p> <pre> firstThree ← digits[3] / 10 lastDigit ← digits[3] MOD 10 sum ← sumDigits(digits[0]) + sumDigits([1]) + sumDigits(digits[2]) + sumDigits(firstThree) isValid ← (sum MOD 10) = lastDigit </pre>	<pre> /* This method validates the credit card number */ private static boolean valid(int [] digits) { int firstThree; // first three digits of last group int lastDigit; // last digit of credit card number int sum; // sum of first 15 digits boolean isValid; // result: true if number is valid // find the first 3 digits pf the last group firstThree = digits[3] / 10; // find the last digit of the number lastDigit = digits[3] % 10; // find the sum of the first 15 digits sum = DigitsLib.sumDigits(digits[0]) + DigitsLib.sumDigits(digits[1]) + DigitsLib.sumDigits(digits[2]) + DigitsLib.sumDigits(firstThree); // determines the validity isValid = (sum % 10 == lastDigit); return isValid; } </pre>

Algorithm Model	Java
DigitsLib Class (DigitsLib.java)	
<p>GIVENS: digits (reference to array of digits) n (number of elements in the array) RESULT: flag (TRUE if first digit is zero, FALSE otherwise) INTERMEDIATE: (none) HEADER: flag ← isZero(digits, n) BODY: flag ← (digits[0] = 0)</p> <p>2nd version: flag ← (digits[0] = 0) AND (digits[1] = 0) AND (digits[2] = 0) AND (digits[3] = 0)</p>	<pre>// first version: only the first digits need to be 0 public static boolean isZero(int [] digits) { boolean flag; // result flag = digits[0] == 0; return(flag); } //second version: all 16 digits need to be 0 public static boolean isZero(int [] digits) { return ((digits[0] == 0) && (digits[1] == 0) && (digits[2] == 0) && (digits[3] == 0)); }</pre>
<p>GIVENS: (none) ITERMEDIATE: RESULT: intArray (reference to integer array) n (number of elements in the array) HEADER: intArray, n ← readDigits() BODY: printLine("Please input the credit card number as four ") printLine("numbers of four digits, separated by spaces;") printLine("or press 0 to finish.") intArray,n ← readIntLine()</p>	<pre>/* This method asks the user to input a credit card number as 4 integers, that will be placed in an array. This method calls readIntLine() from the class ITI1120 to read the array in integers. */ public static int [] readDigits() { int [] intArray; // reference to array System.out.println ("Please input the credit card number as four "); System.out.println ("numbers of four digits, separated by spaces;"); System.out.println("or press 0 to finish."); intArray = ITI1120.readIntLine(); return intArray; }</pre>

Algorithm Model	Java
<p>GIVENS: x (integer number) RESULT: sum (sum of all digits in the integer number) INTERMEDIATE: HEADER: $sum \leftarrow sumDigits(x)$ BODY:</p>  <pre> graph TD Start(()) --> Init[sum ← 0] Init --> Cond{x ≠ 0} Cond -- true --> Body["sum ← sum + (x MOD 10) x ← x/10"] Body --> Cond Cond -- false --> Exit(()) </pre>	<p><i>// Returns the sum of the digits of a number x</i></p> <pre> public static int sumDigits(int x) { int sum; // result - sum of digits // Body sum = 0; while (x != 0) { sum = sum + x % 10; x = x / 10; } return sum; } </pre>