Christopher Bussen

CPS 150 02 – Algorithms and Programming 1

Long Week Project 4

12/6/2020

**Problem 1 Algorithm**

1. Start the program
2. Import the scanner
3. Prompt the user to enter the length of the subarray and declare an int variable for this value
4. Declare an int array variable for the array given in the problem
5. Create a separate method to find the starting index of the subarray of given length with the maximum average – this method should take in an int (represents the length of the subarray) and an int array and return an int value
6. Check to make sure the length of the subarray is less than the length of the actual array – if not, return -1
7. Declare a new int array for the cumulative sum – should have the same length as the original array
8. Use a for loop to fill the cumulative sum array by adding the previous value in the cumulative sum array and the current value in the original array
9. Outside of the for loop, declare an int variable for the maximum sum – set it equal to the component of the cumulative sum array at index length – 1
10. Declare an int variable for the end index of the maximum subarray – set it equal to length of subarray - 1
11. Use a for loop to find the sum of other subarrays
12. In the for loop, declare an int variable for the new sum
13. If the new sum is greater than the max sum, make the new sum the max sum update the end index of the subarray
14. Return the starting index of the maximum subarray – should be equal to the end index – the length of the subarray + 1
15. End the method that finds the starting index and return to the main method
16. Call the method that determines the starting index of the maximum subarray and declare an int variable for this value
17. Declare a double variable for the sum of the subarray
18. Use a for loop that starts at the subarray’s starting index to calculate and update the sum of the subarray
19. Declare a double variable for the maximum average of the array – should be equal to the sum of the subarray / length of the subarray
20. Print the starting index of the subarray and the maximum average of the array
21. End the main method
22. End the program

**Problem 1 Running Screenshot**

**Text

Description automatically generated**

**Problem 1 Code**

/\*

Christopher Bussen

CPS 150 02

Long Week Project 4

MaximumAverageOfSubarray: number ; number number

program takes in an int from the user that represents the length of the subarray

and then calculates and prints the starting index of the maximum average and the

maximum average of the subarray of the given length within the array

array = {1, 12, -5, -6, 50, 3}

ex1: user inputs 4 - program outputs 1, 12.75

ex2: user inputs 2 - program outputs 4, 26.5

ex3: user inputs 3 - program outputs 3, 15.667

ex4: user inputs 1.5 - program outputs error

ex5: user inputs x - program outputs error

\*/

import java.util.Scanner;

public class MaximumAverageOfSubarray {

public static void main(String[] args) {

//import scanner

Scanner input = new Scanner(System.in);

//prompt the user to enter the length of the subarray and declare an int variable for this value

System.out.print("Enter the length of the subarray: ");

int length = input.nextInt();

//declare an int array variable for the array given in the problem

int[] arr = {1, 12, -5, -6, 50, 3};

//call the method that determines the starting index and declare an int variable for this value

int startIndex = calculateStartIndex(length, arr);

//declare a double variable for sum of the subarray

double sumSubarray = 0;

//use for loop to calculate the sum of the subarray

for(int i = startIndex; i < startIndex + length; i++){

sumSubarray = sumSubarray + arr[i];

}

//declare a double variable for the max average of the array

double maxAverage = sumSubarray / length;

//print the starting index and the maximum average

System.out.println("The subarray starts at index " + startIndex);

System.out.println("The maximum average of the subarray is " + maxAverage);

}

//create a separate method to find the starting index of subarray with maximum average and length k - takes in an int and an int array, returns an int

public static int calculateStartIndex(int x, int[] a) {

//check to make sure given length of subarray (x) is less than length of the array (a) - if not, return -1

if (x > a.length) {

return -1;

}

//declare a new int array for the sum - same length as a

int[] sum = new int[a.length];

//use a for loop to fill the sum array

for (int i = 1; i < a.length; i++) {

sum[i] = sum[i - 1] + a[i];

}

//declare an int variable for the max sum - initialize as xth component in array

int maxSum = sum[x - 1];

//declare an int variable for the end position of the max sum - initialize as x-1

int endIndex = x - 1;

//use for loop to find the sum of other subarrays

for (int j = x; j < a.length; j++) {

//declare an int variable to for the sum of new subarray

int newSum = sum[j] - sum[j - x];

//if the new sum is greater than the max sum, make the new sum the max sum and update the end index

if (newSum > maxSum) {

maxSum = newSum;

endIndex = j;

}

}

//return the starting index of the maximum array - equal to end index - length of subarray + 1

return endIndex - x + 1;

}

}

**Problem 2 Algorithm**

NOTE: I know I could’ve easily done this by making the letterGradeToNumericGrade method a long series of if/else if statements for each grade with the final else printing an error message – I didn’t do this because it felt too easy and I wanted to review how to use switch statements before the final exam

1. Start the program
2. Import the scanner
3. Prompt the user to enter their letter grade and declare a string variable for this value
4. Convert the string to uppercase
5. Use a switch statement to check the validity of the entered letter grade – if it’s not a valid letter grade, make the grade variable empty
6. Create a separate method that converts a letter grade to a numeric grade – it should take in a string (the letter grade) and return a double
7. If the string is empty, return -1
8. Declare a double variable for the numeric grade
9. Use a switch statement to assign numeric grades to each base letter grade – A = 4.0, B=3.0, C=2.0, D=1.0, and F=0.0
10. If the letter grade has two characters, add 0.3 to the numeric grade if the second character is a + (unless it’s an A+) and subtract 0.3 from the numeric grade if the second character is a –
11. Return the numeric grade
12. End the method that converts a letter grade to a numeric grade and return to the main method
13. Call the method that converts a letter grade to a numeric grade using the grade string as a parameter – if it returns -1, print an error message, otherwise print the numeric grade
14. End the main method
15. End the program

**Problem 2 Running Screenshot**

**Text

Description automatically generated**

**Problem 2 Code**

/\*

Christopher Bussen

CPS 150 02

Long Week Project 4

ConvertLetterGradeToNumericGrade: string ; number (or possibly string)

program takes in a string from the user for their letter grade and calculates

and prints the numeric grade associated with the letter grade - acceptable letter

grades are: A+, A, A-, B+, B, B-, C+, C, C-, D+, D, D-, F

ex1: user inputs A+ - program outputs 4.0

ex2: user inputs b+ - program outputs 3.3

ex3: user inputs B- - program outputs 2.7

ex4: user inputs D - program outputs 1.0

ex5: user inputs F- - program outputs Invalid letter grade - please try again

ex6: user inputs bad grade - program outputs Invalid letter grade - please try again

\*/

import java.util.Scanner;

public class ConvertLetterGradeToNumericGrade {

public static void main(String [] args) {

//import scanner

Scanner input = new Scanner(System.in);

//prompt the user to enter their letter grade and declare a string variable for it

System.out.print("Enter your letter grade: ");

String grade = input.nextLine();

//convert entered grade to uppercase

grade = grade.toUpperCase();

//check validity of letter grade entered using switch statement - if letter grade is invalid, make string empty

switch (grade) {

case "A+", "A", "A-", "B+", "B", "B-", "C+", "C", "C-", "D+", "D", "D-", "F":

break;

default:

grade = "";

}

//call method that converts letter grade to numeric grade using grade as parameter - if numeric grade is -1, print error message - otherwise, print numeric grade

if(letterGradeToNumericGrade(grade) == -1){

System.out.println("Invalid letter grade - please try again.");

}

else{

System.out.println("The numeric grade is " + letterGradeToNumericGrade(grade));

}

}

//create a separate method that converts a letter grade to a numeric grade - takes in a string and returns a double

public static double letterGradeToNumericGrade(String a){

//if string is empty, return -1

if(a.length() == 0){

return -1;

}

//declare a double variable for numeric grade

double numericGrade = 0;

//use a switch statement to assign numeric grades for each base letter grade

switch (a.charAt(0)){

case 'A':

numericGrade = 4.0;

break;

case 'B':

numericGrade = 3.0;

break;

case 'C':

numericGrade = 2.0;

break;

case 'D':

numericGrade = 1.0;

break;

case 'F':

numericGrade = 0.0;

break;

}

//if the grade has 2 characters, add 0.3 for a + (unless it's an A+) and subtract 0.3 for a -

if(a.length() == 2){

if(a.charAt(1) == '+' && a.charAt(0) != 'A'){

numericGrade = numericGrade + 0.3;

}

else if(a.charAt(1) == '-'){

numericGrade = numericGrade - 0.3;

}

}

//return the numeric grade

return numericGrade;

}

}

**Problem 3 Algorithm**

1. Start the program
2. Import the scanner
3. Prompt the user to enter their numeric grade and declare a double variable for it
4. Create a separate method that converts a numeric grade to a letter grade – should take in a double (numeric grade) and return a string
5. Declare a string variable for the letter grade
6. Use if statements to change the letter grade variable to the correct letter grade closest to the numeric grade and round up on ties – for example, if the numeric grade is a 2.85, the closest letter grade is a B
7. If the numeric grade is greater than 4 or less than 0, leave the letter grade string empty
8. Return the letter grade string
9. End the method that converts a numeric grade to a letter grade and return to the main method
10. Call the method that converts a numeric grade to a letter grade using the numeric grade as a parameter – if it returns an empty string, tell the user to enter a numeric grade between 0 and 4.0
11. Otherwise, print the letter grade closest to the numeric grade
12. End the main method
13. End the program

**Problem 3 Running Screenshot**

**Text

Description automatically generated**

**Problem 3 Code**

/\*

Christopher Bussen

CPS 150 02

Long Week Project 4

ConvertNumberGradeToLetterGrade: number ; string

program takes in a number between 0 and 4 from the user for their numeric grade and determines and prints the letter grade closest to the numeric grade -- NOTE: A 4.0 corresponds to an

A - there are no A+'s when converting from numeric grade to letter grade

ex1: user inputs 2.85 - program outputs B

ex2: user inputs 4 - program outputs A

ex3: user inputs 3.34 - program outputs B+

ex4: user inputs 1.841 - program outputs C-

ex5: user inputs x - program outputs error

ex6: user inputs 5.63 - program outputs Enter a numeric grade between 0 and 4.0

ex7: user inputs -0.5 - program outputs Enter a numeric grade between 0 and 4.0

\*/

import java.util.Scanner;

public class ConvertNumberGradeToLetterGrade {

public static void main(String [] args){

//import scanner

Scanner input = new Scanner(System.in);

//prompt the user to enter their numeric grade and declare a double variable for it

System.out.print("Enter your numeric grade: ");

double grade = input.nextDouble();

//call the method that converts a numeric grade to a letter grade using the user input grade as a parameter

//if the returned string is empty, give user error message - otherwise print the letter grade

if(numberGradeToLetterGrade(grade).equals("")){

System.out.println("Enter a numeric grade between 0 and 4.0");

}

else{

System.out.println("The letter grade is " + numberGradeToLetterGrade(grade));

}

}

//create a separate method that converts a numeric grade to a letter grade - takes in a double and returns a string

public static String numberGradeToLetterGrade(double a){

//declare a string variable for the letter grade

String letterGrade = "";

//use if statements to change the letter grade variable to the correct value associated with the numeric grade

//if the numeric grade is greater than 4 or less than 0, leave the string empty

if(a > 4){

}

else if(a >= 3.85){

letterGrade = "A";

}

else if(a >= 3.50){

letterGrade = "A-";

}

else if(a >= 3.15){

letterGrade = "B+";

}

else if(a >= 2.85){

letterGrade = "B";

}

else if(a >= 2.50){

letterGrade = "B-";

}

else if(a >= 2.15){

letterGrade = "C+";

}

else if(a >= 1.85){

letterGrade = "C";

}

else if(a >= 1.50){

letterGrade = "C-";

}

else if(a >= 1.15){

letterGrade = "D+";

}

else if(a >= 0.85){

letterGrade = "D";

}

else if(a >= 0.35){

letterGrade = "D-";

}

else if(a >= 0){

letterGrade = "F";

}

else{

}

//return the letter grade

return letterGrade;

}

}