**Baku Higher Oil School**

**Process Automation Engineering Department**

**Programming in C**

**Laboratory 2 – Control Flow**

**P.S** Add comment for each task; submit the file in LMS before the deadline.

1. Identify and correct the errors in each of the following. [Note: There may be more than one error in each piece of code.]
2. **if** ( sales => **5000** )

puts( **"Sales are greater than or equal to $5000"** )

**else**

puts( **"Sales are less than $5000** )

1. **int** x = **1**, product = **0**;

**while** ( x <= **10** ); {

product \*= x;

++x;

}

1. **While** ( x <= **100** )

total =+ x;

++x;

1. **while** ( y < **10** ) {

printf( **"%d\n"**, y );

}

1. What does the following program print?

Text

Description automatically generated

1. Obtain a series of positive numbers from the keyboard and determine and display their average. Assume that the user types the sentinel value -1 to indicate “end of data entry.”
2. Write a program to check whether an alphabet entered by the user is a vowel or a consonant.
3. Write a program to check whether a character entered by the user is an alphabet or not.
4. Write a program to calculate the LCM (Lowest common multiple) of two numbers entered by the user.
5. Write a program to calculate the power of a number using while.
6. Write a program to check whether the number entered by the user is a palindrome or not.
7. **(Mortgage Calculator)** Develop a C program to calculate the interest accrued on a bank customer's mortgage. For each customer, the following facts are available:

* the account number
* the mortgage amount
* the mortgage term
* the interest rate

The program should input each fact, calculate the total interest payable ( = mortgage amount× interest rate × mortgage term), and add it to the mortgage amount to get the total amount payable. It should calculate the required monthly payment by dividing the total amount payable by the number of months in the mortgage term. The program should display the required monthly payment rounded off to the nearest dollar. The program should process each customer's account at a time. Here is a sample input/ output dialog:

|  |
| --- |
| Enter account number (-1 to end): 100  Enter mortgage amount (in dollars): 6500  Enter mortgage term (in years): 3  Enter interest rate (as a decimal): 0.075  The monthly payable interest $ 221  Enter account number (-1 to end): 200  Enter mortgage amount (in dollars): 12000  Enter mortgage term (in years): 10  Enter interest rate (as a decimal): 0.045  The monthly payable interest is: $ 145  Enter account number (-1 to end): -1 |

1. ***(Find the Largest Number)*** The process of finding the largest number (i.e., the maximum of a group of numbers) is used frequently in computer applications. For example, a program that determines the winner of a sales contest would input the number of units sold by each salesperson. The salesperson who sells the most units wins the contest. Write a pseudocode program and then a program that inputs a series of 10 non-negative numbers and determines and prints the largest of the numbers. [*Hint:* Your program should use three variables as shown below.]

|  |  |
| --- | --- |
| Counter: | A counter to count to 10 (i.e., to keep track of how many numbers have been input and to determine when all 10 numbers have been processed) |
| Number: | The current number input to the program |
| Largest: | The largest number found so far |

1. ***(Tabular Output)*** Write a program that uses looping to print the following table of values. Use the tab escape sequence, \t, in the printf statement to separate the columns with tabs.

Shape, rectangle

Description automatically generated

1. What does the following program print?

Graphical user interface, text

Description automatically generated

1. What does the following program print?

Text

Description automatically generated

1. **(Armstrong Numbers)** Armstrong numbers are numbers that are equal to the sum of their digits raised to power of the number of digits in them. The number 153, for example, equals . Thus, it is an Armstrong number. Write a program to display all three-digit Armstrong numbers.
2. **(Counting 9s)** Write a program that reads an integer (5 digits or fewer) and determines and prints how many digits in the integer are 9s.:
3. **(Factorial)** The factorial of a nonnegative integer n is written n! (pronounced “n factorial”) and is defined as follows:

and

For example, 5! = 5 · 4 · 3 · 2 · 1, which is 120.

a) Write a program that reads a nonnegative integer and computes and prints its factorial.

b) Write a program that estimates the value of the mathematical constant e by using the formula:

c) Write a program that computes the value of ex by using the formula

1. **(Population-Growth Calculator)** Use the web to determine the current world population and the annual world population growth rate. Write an application that inputs these values, then displays the estimated world population after one, two, three, four and five years.
2. **(Enforcing Privacy with Cryptography)** The explosive growth of Internet communications and data storage on Internet-connected computers has greatly increased privacy concerns. The field of cryptography is concerned with coding data to make it difficult (and hopefully—with the most advanced schemes—impossible) for unauthorized users to read. In this exercise you’ll investigate a simple scheme for encrypting and decrypting data. A company that wants to send data over the Internet has asked you to write a program that will encrypt it so that it may be transmitted more securely. All the data is transmitted as four-digit integers. Your application should read a four-digit integer entered by the user and encrypt it as follows: Replace each digit with the result of adding 7 to the digit and getting the remainder after dividing the new value by 10. Then swap the first digit with the third, and swap the second digit with the fourth. Then print the encrypted integer. Write a separate application that inputs an encrypted four-digit integer and decrypts it (by reversing the encryption scheme) to form the original number.

**LOOPS**

1. Identify and correct the errors in each of the following. [Note: There may be more than one error in each piece of code.]
2. x = **1**;

**while** (x <= **10**);

++x;

}

1. **for** (**double** y = **.1**; y != **1.0**; y += **.1**) {

printf(**"%f\n"**, y);

}

1. **switch** (n) {

**case 1**:

puts(**"The number is 1"**);

**case 2**:

puts(**"The number is 2"**);

**break**;

**default**:

puts(**"The number is not 1 or 2"**);

**break**;

}

1. The following code should print the values 1 to 10.

n = **1**;

**while** (n < **10**) {

printf(**"%d "**, n++);

}

1. What does the following program print?

Text, letter

Description automatically generated

1. Print numbers from 1 to 10 using for, while and do while loop.
2. **(Factorials)** The factorial function is used frequently in probability problems. The factorial of a positive integer n (written n! and pronounced “n factorial”) is equal to the product of the positive integers from 1 to n. Write a program that evaluates the factorials of the integers from 1 to 5. Print the results in tabular format.
3. **(Prime Numbers)** Write a program to calculate and print a list of all prime numbers from 1 to 100.
4. Display the number in reverse order.
5. Print even numbers from 1 to N using while loop

**Input:**

Enter value of N: 10

**Output:**

Even numbers from 1 to 10:

2 4 6 8 10.

1. Input week number (1-7) and print day of week name using switch case.

**Input:**

Input week number (1-7): 2

**Output:**

Tuesday.

1. Input two numbers from user and find maximum between two numbers using switch case.

**Input:**

Input first number: 15

Input second number: 8

**Output:**

Maximum: 15

1. Input number from user and check whether the number is even or odd using switch case.

**Input:**

Input number: 12

**Output:**

Even number

1. Write a statement or a set of statements to accomplish each of the following tasks:
2. Sum the odd integers between 1 and 99 using a for statement. Use the unsigned integer variables sum and count.
3. Print the value 333.546372 in a field width of 15 characters with precisions of 1, 2, 3, 4 and 5. Left justify the output. What are the five values that print?
4. Calculate the value of 2.5 raised to the power of 3 using the pow function. Print the result with a precision of 2 in a field width of 10 positions. What is the value that prints? (use math.h library)
5. Print the integers from 1 to 20 using a while loop and the counter variable x. Print only five integers per line. [Hint: Use the calculation x % 5. When the value of this is 0, print a newline character, otherwise print a tab character.]
6. Repeat Exercise 4.3(d) using a for statement.

1. **(Conversion Celsius to Fahrenheit)** Write a program that converts temperatures from 30 C to 50 C to the Fahrenheit scale. The program should print a table displaying temperatures in the two scales side by side. [Hint]
2. **(Calculating the Sum of Multiples)** Write a program to calculate and print the sum of all multiples of 7 from 1 to 100.

***Optional questions - advanced version***

1. **(Table of Decimal, Binary, Octal and Hexadecimal Equivalents)** Write a program that prints a table of the binary, octal and hexadecimal equivalents of the decimal numbers in the range 1 through 256. [Note: You can display an integer as an octal or hexadecimal value with the conversion specifiers %o and %X, respectively.]
2. **(Calculating the Value of π)** Calculate the value of π from the infinite series

Print a table that shows the value of π approximated by one term of this series, by two terms, by three terms, and so on. How many terms of this series do you have to use before you first get 3.14? 3.141? 3.1415? 3.14159?