



Problem Set 01

Please write your solutions in A4/Letter size paper and provide the handwritten or printed hardcopy in the department. Aim for concise solutions; convoluted and obtuse descriptions might receive low marks, even when they are correct.

This assignment is meant to be an evaluation of your individual understanding coming into the course and should be completed without collaboration or outside help

Submission Deadline: 22 February, 2024

*(Practice all but submit any 150 points; At least 30 points from each section.
Each question contributes 10 points)*

Section 1. [100 points] Questions – Answers

[Provide concise responses without unnecessary elaboration or additional details]

- (a) What are the major components of a C program? What significance is attached to the name main?
- (b) Summarize the rules for naming identifiers. In your answer, specify the following :
What is the purpose of identifiers? Are uppercase letters equivalent to lowercase letters? Can digits be included in an identifier name? Can any special characters be included? How many characters can be included in an identifier name? Are all of these characters equally significant?
- (c) In what general category do the #define and #include statements fall? What are the differences between a symbolic constant and constant define with **const** keyword.
- (d) Can you discuss the trade-offs between using an interpreted language versus a compiled language for a project? What factors would you consider when making this decision?
- (e) What is the purpose of the modulus operator % in C? Differentiate between == and = in C. Explain the difference between the ++i and i++ increment operators.
- (f) What is the purpose of the do - while statement? How does it differ from the while statement? What is the minimum number of times that a do - while loop can be executed? Compare with a while loop and explain the reasons for the differences.
- (g) What is the purpose of the **switch** statement? How does this statement differ from **if else** other statements? Compare the use of the switch statement with the use of nested if- else statements. Which is more convenient?
- (h) What is the purpose of the break statement? Within which control statements can the break statement be included? Suppose a break statement is included within the innermost of several nested control statements. What happens when the break statement is executed?
- (i) What is the purpose of the continue statement? Within which control statements can the continue statement be included? Compare with the break statement.
- (j) Define entry control and exit control loop with their syntax. Discuss their similarity and dissimilarity. Explain the concept of an infinite loop and how to avoid it.



Rubric:

- **Incorrect Answer:** Deduct 2 points for each incorrect answer.
- **Correct Answer:** Award 5 points for each correct answer.
- **Explanation:** Award 3 points for providing a clear and relevant explanation along with the correct answer.
- **Partial Credit:** Partial credit may be granted based on demonstrated understanding, even if the solution does not yield the correct result.

Section 2. [160 points] Problems – Solutions

- (a) Write a simple C program to find the largest number among three using if-else statements
- (b) Write a simple C program to sort three numbers in ascending order using if-else statements
- (c) Let BAUST admission test is subjected to the following conditions :
- Marks in Mathematics ≥ 60
 - Marks in Physics ≥ 50
 - Marks in Chemistry ≥ 40
 - Total marks in all three subjects ≥ 200 or Total in Mathematics and Physics ≥ 150

Given the marks in three subjects, write a program to process the application to check the eligible candidates.

- (d) In any n credit theory course of BAUST you have to attend in n CTs and of them best n-1 CTs will be counted. Now write a program that shows the average of best n-1 CTs marks out of n CTs.

Sample input:

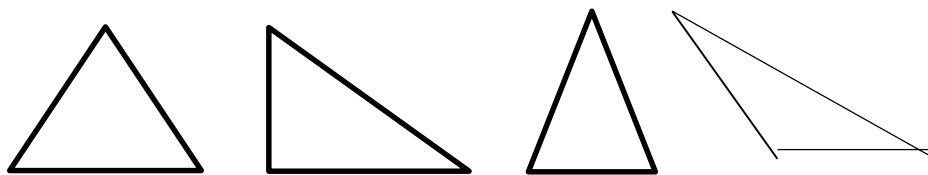
Enter the value of n: 4

- 1st CT Marks: 15
- 2nd CT Marks: 10
- 3rd CT Marks: 15
- 4th CT Marks: 15

Sample Output:

Average of best 3 CT marks: 15

- (e) We know, there is a rule for valid triangle and there are various types of triangles in terms of their angle and sides' length. Now observe some of the following triangle shapes.

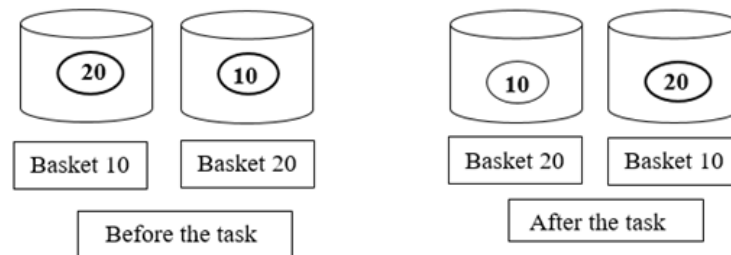


- Present a C program to check whether a triangle is valid or not if sides are given.
 - Compile a C program to check whether a triangle is equilateral, scalene or isosceles.
- (f) Write a program in C that will ask the user to enter the total shopping amount purchased in the shop and then apply the discount as per the following discount criteria, then find and print the final amount that has to be paid by the customer after subtracting the discount amount:
- If the user has purchased something that costs from \$101 to \$200, the discount will be 5%.
 - If the user has purchased something that costs from \$201 to \$400, the discount will be 10%.



- If the user has purchased something that costs from \$401 to \$800, the discount will be 20%.
- And if the user has purchased something that costs more than \$800, the total discount that will be applied to the whole purchase amount is 25%.

(g) Consider the following scenarios, you are given two baskets, one basket is labelled as 10 number and another one is labelled as 20 number. In label 10 basket, there is a ball which is labelled as number 20, and in another basket, label 10 number ball is present. Now, your task is to keep those two balls in those two baskets so that the baskets and balls label number match.



- Write a C program for your assigned task.
- Write an alternative C program for this assigned task.

Note: You can rename your basket label number maintaining the rules but the balls label number are strictly maintained.

(h) Write a program that takes a character as input and prints out one of the following sentences as appropriate:

- Vowel in capital letter
- Vowel in small letter
- Consonant in capital letter
- Consonant in small letter
- Neither a vowel nor a consonant

(i) Leap years are years where an extra day is added to the end of the shortest month, February. This so-called intercalary day, February 29, is commonly referred to as leap day. Leap years have 366 days instead of the usual 365 days and occur almost every four years.

A leap year is exactly divisible by 4 except for century years (years ending with 00). The century year is a leap year only if it is perfectly divisible by 400. For example,

- 1999 is not a leap year
- 2000 is a leap year
- 2004 is a leap year

Now, write a C Program to check whether the year entered by the user is a leap year or not.

(j) Write a simple C program to reverse an integer.

Sample input: 123

Sample Output: 321

(k) A palindromic number (also known as a numeral palindrome or a numeric palindrome) is a number (such as 16461) that remains the same when its digits are reversed. In other words, it has relational symmetry across a vertical axis. Now, write a C program to check whether a number is palindrome or not.

(l) Write an appropriate control structure that will examine the value of a floating-point variable called temp and print one of the following messages, depending on the value assigned to temp.

- ICE, if the value of temp is less than 0.



- ii. WATER, if the value of temp lies between 0 and 100.
- iii. STEAM, if the value of temp exceeds 100.

Can a switch statement be used in this instance?

(m) Write a switch statement that will examine the value of a char-type variable called **color** and print one of the following messages, depending on the character assigned to color.

- i. RED, if either r or R is assigned to color,
- ii. GREEN, if either g or G is assigned to color,
- iii. BLUE, if either b or B is assigned to color,
- iv. BLACK, if color is assigned any other character.

(n) Write an appropriate control structure that will examine the value of a floating-point variable called temp and print one of the following messages, depending on the value assigned to temp.

- i. ICE, if the value of temp is less than 0.
- ii. WATER, if the value of temp lies between 0 and 100.
- iii. STEAM, if the value of temp exceeds 100.

Can a switch statement be used in this instance?

(o) Write a loop that will calculate the sum of every third integer, beginning with i = 2 (i.e. calculate the sum $2 + 5 + 8 + 11 + \dots$) for all values of that are less than 100. Write the loop in three different ways.

- a. Using a *while* statement.
- b. Using a *do - while* statement.
- c. Using a *for* statement.

(p) Implement a C program to calculate the factorial of a given number using a recursive function. Test the program with various input values.

Sample input: 5

Sample Output: 120

(q) A prime number is a whole number greater than 1 whose only factors are 1 and itself. A factor is a whole number that can be divided evenly into another number. The first few prime numbers are 2, 3, 5, 7, 11, 13, 17, 19, 23 and 29. Numbers that have more than two factors are called composite numbers. Now, write a C Program that will print all the prime number up to a positive integer n.

(r) An Armstrong number is one whose sum of digits raised to the power three equals the number itself. 371, for example, is an Armstrong number because $3^3 + 7^3 + 1^3 = 371$. Now, write a C Program to Display Armstrong Number between Two Intervals.

(s) The Fibonacci sequence is a sequence where the next term is the sum of the previous two terms. The first two terms of the Fibonacci sequence are 0 followed by 1. Now, write a C Program to display the Fibonacci sequence of first n numbers where n is a positive integer less than 100.

Rubrics:

- 1 points for indentation
- 1 points for documentation
- 1 points for coherence
- 2 points for logical integrity
- 5 points for correct code
- Partial credit may be awarded



Section 3. [60 points] Print Pyramids and Patterns: Write C Program to display the following Pyramids or Patterns

<pre> * ** *** **** ***** ***** </pre>	<pre> 1 2 3 3 4 5 4 3 4 5 6 7 6 5 4 5 6 7 8 9 8 7 6 5 </pre>	<pre> ***** ***** ***** *** * </pre>
a) Stars Pyramid	b) Numbers Pyramid	c) Inverted Stars Pyramid
<pre> 1 1 1 1 2 1 1 3 3 1 1 4 6 4 1 1 5 10 10 5 1 </pre>	<pre> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 </pre>	<pre> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 </pre>
c) Pascal's Triangle	d) Floyd's Triangle	e) Floyd's Pyramid

Rubrics:

- 5 points for each correct pyramid or pattern
- Partial credit may be awarded

Section 4. [80 points] Describe the output that will be generated by each of the following C programs. [Note the similarities in the programs that are shown across from each other.]

a.	<pre> #include <stdio.h> int main() { int i = 0, x = 0; while (i < 20) { if (i % 5 == 0) { x += i; x++; printf("%d ", x); } ++i; } printf("\nx = %d", x); return 0; } </pre>	b.	<pre> #include <stdio.h> int main() { int i = 0, x = 0; do { if (i % 5 == 0) { x += i; x++; printf("%d ", x); } ++i; } while (i < 20); printf("\nx = %d", x); return 0; } </pre>
c.	<pre> #include <stdio.h> int main() { int i = 0, x = 0; for (i = 1; i < 10; i *= 2) { x++; if (i % 2 == 1) { printf("%d ", x); } x += i; } printf("\nx = %d", x); return 0; } </pre>	d.	<pre> #include <stdio.h> int main() { int i = 0, x = 0; for (i = 1; i < 10; ++i) { if (i % 2 == 1) { x += i; } else { x--; } printf("%d ", x); continue; break; } } </pre>



			<pre>printf("\nx = %d", x); return 0; }</pre>
g.	<pre>#include <stdio.h> int main() { int i, j, x = 0; for (i = 0; i < 5; ++i) { for (j = 0; j < i; ++j) { x += (i + j - 1); printf("%d ", x); break; } printf("\nx = %d\n", x); } return 0; }</pre>	h.	<pre>#include <stdio.h> int main(){ int i, j, k, x = 0; for (i = 0; i < 5; ++i) for (j = 0; j < i; ++j){ k = (i + j - 1); if (k % 2 == 0) x += k; else if (k % 3 == 0) x += k - 2; printf("%d ", x); } printf("\n x = %d ",x) ; return 0; }</pre>
i.	<pre>#include <stdio.h> int main() { int i, j, k, x = 0; for (i = 0; i < 5; ++i) { for (j = 0; j < i; ++j) { switch (i + j - 1) { case -1: case 0: x += 1; break; case 1: case 2: case 3: x += 2; break; default: x += 3; } printf("%d ", x); } printf("\nx = %d\n", x); } return 0; }</pre>	j.	<pre>#include <stdio.h> int main() { int i, j, k, x = 0; for (i = 0; i < 5; ++i) { for (j = 0; j < i; ++j) { switch (i + j - 1) { case -1: case 0: x += 1; break; case 1: case 2: case 3: x += 2; default: x += 3; } printf("%d ", x); } printf("\nx = %d\n", x); } return 0; }</pre>

Rubric:

- -1 point for each wrong answer
- 5 points for correct answer
- 5 points for description
- Partial credit may be awarded if there is work shown of a correct approach that does not yield the correct solution.