Orchid International College

Bijaychowk, Gaushala

Department Of Information Technology



LAB MANUAL

C programming

BIM, BCA, BCS.CSIT

# Objective

# To gain Knowledge on C programming Language

# Student able to Work on code block

# Student will learn about decision making, iterative, statements in c

# Able to handle files with C

# Use graphs in c

# Hardware and Software Required:

# A working computer system with any operating system.

# Codeblocks

# 

# Outcomes:

**C programming**

1. Introduction To C programming
2. Simple Programs
3. Decision making
4. Iterative Statements
5. Working With Array
6. Function
7. String handling
8. Pointer and Structure
9. union
10. File Handling
11. Graphics

**LAB-2 (Simple Programs)**

1. WAP to find area of a circle.

#include <stdio.h>

#include<conio.h>

int main() {

float radius, area;

// Ask user to enter the radius

printf("Enter the radius of the circle: ");

scanf("%f", &radius);

// Calculate the area

area = 3.14 \* radius \* radius;

// Print the area

printf("The area of the circle is: %f\n", area);

return 0;

}

2. WAP to find volume of a cylinder.

3. WAP to input temperature in fahrenheit and convert into celsius.

[c=5/9(f-32) ]

4. WAP to input a number and find its square root.

5. WAP to input 3 numbers and find their average.

6. WAP to find simple interest.

7. WAP to input two numbers find the cube of first number, square of second number and then add those results to find the new result.

**LAB-3 (Decision Making Statements):** Decision-making statements in C are used to control the flow of execution based on certain conditions. These statements allow the program to make decisions and execute certain sections of code while skipping others, based on the evaluation of a condition. The primary decision-making statements in C are if, if-else, else if, and switch.

1. WAP to check odd/even.

#include <stdio.h>

#include<conio.h>

int main() {

int number;

// Ask the user to enter a number

printf("Enter an integer: ");

scanf("%d", &number);

// Check if the number is odd or even

if (number % 2 == 0) {

printf("%d is even.\n", number);

} else {

printf("%d is odd.\n", number);

}

return 0;

}

2. WAP to input 3 numbers and find the middle number.

3. WAP to input electricity units and find the billing amount.

for first 20 units - Rs 0/unit

for next 100 units - Rs 12/unit

for next 100 units - Rs 11/unit

for above 220 units - Rs 10/unit

4. WAP a program to check vowel/consonant.

5. WAP to input two numbers and then input the operator(+,-,\*,/) and find the result using switch.

6. Create a menu driven program to add/subtract/multiply/divide two numbers using switch.

**LAB-4 (Iterative Statements):** Iterative statements, also known as loops, are used in programming to repeatedly execute a block of code as long as a specified condition is true. C provides several types of loops, each with its specific use case. For, while, do-while are example of Iterative statements in c

1. WAP to display all numbers 100 to 200 that end with 0 or 1.

#include <stdio.h>

#include<conio.h>

int main() {

int i;

// Loop through numbers from 100 to 200

for (i = 100; i <= 200; i++) {

// Check if the number ends with 0 or 1

if (i % 10 == 0 || i % 10 == 1) {

printf("%d\n", i);

}

}

return 0;

}

2. WAP to display the following series:

a. \* b. \* c. 5 d. 1 e. \*\*\*\*\*\*\* f. 1

\*\* \*\*\* 45 212 \*\*\*\*\* 10

\*\*\* \*\*\*\*\* 345 32123 \*\*\* 101

\*\*\*\* \*\*\*\*\*\*\* 2345 4321234 \* 1010

\*\*\*\*\* 12345 543212345 10101

3. WAP to input a number and count the number of digits.

4. WAP to display all prime numbers between 100 and 200.

5. WAP to display all palindrome numbers from 100 to 500.

**LAB-5 (Array):** An array in C is a collection of elements of the same data type stored in contiguous memory locations. Arrays allow you to store multiple values under a single variable name, making it easy to manage large amounts of data.

1. WAP to input numbers in an array of size 'N' and find the sum of even and odd numbers separately.

#include <stdio.h>

#include<conio.h>

int main() {

int N, i;

int evenSum = 0, oddSum = 0;

// Ask the user for the size of the array

printf("Enter the size of the array: ");

scanf("%d", &N);

int arr[N]; // Declare the array with size N

// Input elements into the array

printf("Enter %d elements:\n", N);

for (i = 0; i < N; i++) {

scanf("%d", &arr[i]);

}

// Calculate the sum of even and odd numbers

for (i = 0; i < N; i++) {

if (arr[i] % 2 == 0) {

evenSum += arr[i];

} else {

oddSum += arr[i];

}

}

// Output the results

printf("Sum of even numbers: %d\n", evenSum);

printf("Sum of odd numbers: %d\n", oddSum);

return 0;

}

2. WAP to input numbers in an array and find the greatest and smallest number

3. WAP to sort the elements of an array in descending order.

4. WAP to print to find sum of two matrices.

5. WAP to multiply two matrices.

**LAB-6 (Function):** A function in C is a self-contained block of code that performs a specific task. Functions allow you to break down complex programs into smaller, manageable, and reusable parts.

1. Create a function void check(int n) that checks whether given number is odd or even.

#include <stdio.h>

#include<conio.h>

// Function to check whether the number is odd or even

void check(int n) {

if (n % 2 == 0) {

printf("%d is even.\n", n);

} else {

printf("%d is odd.\n", n);

}

}

int main() {

int number;

// Ask the user to enter a number

printf("Enter an integer: ");

scanf("%d", &number);

// Call the check function

check(number);

return 0;

}

2. Create a function int small(int a,int b) that finds the smaller number between two numbers.

3. Create a function that takes one integer argument and finds the sum of its digits.

4. Create a function that takes an int array as argument and returns the smallest value in the array.

5. WAP to find the factorial of a number using recursive function.

6. WAP to display the first 10 numbers in a fibonacci series using recursive function.

**LAB-7(String Handling):** In C, strings are represented as arrays of characters terminated by a null character ('\0'). This null character signifies the end of the string. For example, "Hello" is represented as {'H', 'e', 'l', 'l', 'o', '\0'}.

1. WAP to input a word and count the number of vowels in it.

#include <stdio.h>

#include <conio.h>

int main() {

char word[100];

int i, count = 0;

// Ask the user to enter a word

printf("Enter a word: ");

scanf("%s", word);

// Traverse each character in the word

for (i = 0; word[i] != '\0'; i++) {

// Convert the character to lowercase to simplify the comparison

char ch = word[i];

// Check if the character is a vowel

if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {

count++;

}

}

// Output the total count of vowels

printf("Number of vowels in the word: %d\n", count);

return 0;

}

2. WAP to input 3 words and print the shortest word.

3. WAP to input a word and check whether it is palindrome or not.

4. WAP to display the following series:

N N KATHMANDU

NE NEP KATHMAN

NEP NEPAL KATHM

NEPA KAT

NEPAL K

**LAB-8 (Pointer and Structure):** Pointers in C are variables that store memory addresses. They are powerful tools that allow direct manipulation of memory and facilitate dynamic memory allocation. Pointers are used extensively in C for tasks like passing arguments to functions by reference, implementing data structures like linked lists and trees, and optimizing memory usage in programs.

Structures in C allow programmers to define a custom data type that can hold multiple variables of different types under a single name. This grouping of variables simplifies code organization and improves readability by logically organizing related data. Structures can contain primitive data types like integers and floats, as well as other structures, arrays, and pointers.

1. WAP to find the greater between two numbers by the use of pointer.

#include <stdio.h>

#include<conio.h>

// Function to find the greater number using pointers

int findGreater(int \*num1, int \*num2) {

if (\*num1 > \*num2) {

return \*num1;

} else {

return \*num2;

}

}

int main() {

int a, b, greater;

// Input two numbers

printf("Enter two numbers: ");

scanf("%d %d", &a, &b);

// Find the greater number

greater = findGreater(&a, &b);

// Output the greater number

printf("The greater number is: %d\n", greater);

return 0;

}

2. WAP to show both pass by value and pass by reference.

3. Create a structure to store name, address, salary of a person and display the person details.

4. Create an array of structure to store name, address, salary of 10 persons and display the details of highest paid person.

5. Create an array of structure to store name, address, salary of 5 persons and sort the records in ascending order of name.

**LAB-9 (File Handling):** File handling in C refers to the process of working with files on a computer system using C programming language. It involves tasks such as reading from files, writing to files, and modifying file contents.

1. WAP to copy the content of file "tu.txt" into "university.txt".

#include <stdio.h>

#include<conio.h>

int main() {

FILE \*sourceFile, \*destFile;

char ch;

// Open the source file in read mode

sourceFile = fopen("tu.txt", "r");

if (sourceFile == NULL) {

printf("Unable to open source file.\n");

return 1;

}

// Open the destination file in write mode

destFile = fopen("university.txt", "w");

if (destFile == NULL) {

printf("Unable to open destination file.\n");

fclose(sourceFile);

return 1;

}

// Copy content from source file to destination file

while ((ch = fgetc(sourceFile)) != EOF) {

fputc(ch, destFile);

}

printf("File copied successfully.\n");

// Close the files

fclose(sourceFile);

fclose(destFile);

return 0;

}

2. A data file "number.txt" contains some numbers. WAP to read the numbers and store all even numbers into "even.txt" and all odd numbers into "odd.txt".

3. Write a program to input a line of text store it in a file and then read from file and display its contents.

4. A data file contains name, address, salary of some person. WAP to input any address and show the records matching that address.

5. A data file contains name, address, salary of some person. WAP to increase salary of all persons by 15%.

6. Create a menu driven program to store and display records (name, address, salary of a person)

1. Add Record

2. Display Record

Enter your choice:

**LAB - 10 (Graphics):** Graphics in C refers to the ability to create and manipulate visual elements such as shapes, lines, text, and images on a computer screen using the C programming language. It involves utilizing libraries like graphics.h or OpenGL to render graphical content.

1. WAP to display Tribhuvan University inside a rectangle.

#include <stdio.h>

#include <string.h>

int main() {

const char \*text = "Tribhuvan University";

int length = strlen(text);

int width = length + 4; // 2 spaces on each side of the text

int height = 5; // top border, bottom border, and text row with 1 space padding each

// Print top border

for (int i = 0; i < width; i++) {

printf("\*");

}

printf("\n");

// Print empty line above the text

printf("\*");

for (int i = 0; i < width - 2; i++) {

printf(" ");

}

printf("\*\n");

// Print the text line

printf("\* %s \*\n", text);

// Print empty line below the text

printf("\*");

for (int i = 0; i < width - 2; i++) {

printf(" ");

}

printf("\*\n");

// Print bottom border

for (int i = 0; i < width; i++) {

printf("\*");

}

printf("\n");

return 0;

}

2. WAP to draw a square and a circle and fill a pattern in both of them.

3. WAP to draw input xco,yco,radius and draw a circle.c

4. WAP to draw 10 concentric circles with different colors.