Module 2 – Introduction to Programming

Q1. Research and provide three real-world applications where C programming is extensively used, such as in embedded systems, operating systems, or game development.

* **1. Embedded Systems**  
  C is commonly used in devices like washing machines, microwaves, TVs, and remote controls. These devices have small computers inside them called microcontrollers. C is perfect for programming these because it runs fast and uses less memory.
* *Example:*  
  In a microwave oven, C code controls how long the food is heated and at what power level.
* **2. Operating Systems**  
  Most operating systems are written in C, like Windows, Linux, and UNIX. Operating systems need to be very efficient and close to hardware. C allows programmers to control memory and hardware directly.
* *Example:*  
  The Linux operating system, which powers many servers and Android phones, is mostly written in C.
* **3. Game Development (Game Engines)**  
  C is also used in game engines – the software used to build video games. It helps manage the speed and performance of games.
* *Example:*  
  Popular game engines like Unity and Unreal Engine use C or C++ for coding the logic of games and making them run smoothly.

Q2. Install a C compiler on your system and configure the IDE. Write your first program to print "Hello, World!" and run it.

* #include<stdio.h>  
  int main()

{  
 printf(“Hello, World”);

Return 0;

}

Q3. Write a C program that includes variables, constants, and comments. Declare and use different data types (int, char, float) and display their values.

* #include<stdio.h>

Int main()

{

Int roll\_no =1;

Float marks = 79.2;

Char grad = ‘B’;

Printf(“roll no %d got %f marks and the grad is %c”,roll\_no,marks,grad);

Return 0;

}

Q4. Write a C program that accepts two integers from the user and performs arithmetic, relational, and logical operations on them. Display the results.

* .

#include <stdio.h>

int main()

{

int num1, num2;

printf("Enter first number: ");

scanf("%d", &num1);

printf("Enter second number: ");

scanf("%d", &num2);

printf("\n-- Arithmetic --\n");

printf("Addition: %d\n", num1 + num2);

printf("Subtraction: %d\n", num1 - num2);

printf("Multiplication: %d\n", num1 \* num2);

printf("Division: %d\n", num1 / num2);

printf("Modulus: %d\n", num1 % num2);

printf("\n-- Relational --\n");

printf("Is Equal: %d\n", num1 == num2);

printf("Not Equal: %d\n", num1 != num2);

printf("Greater: %d\n", num1 > num2);

printf("Less: %d\n", num1 < num2);

printf("\n-- Logical --\n");

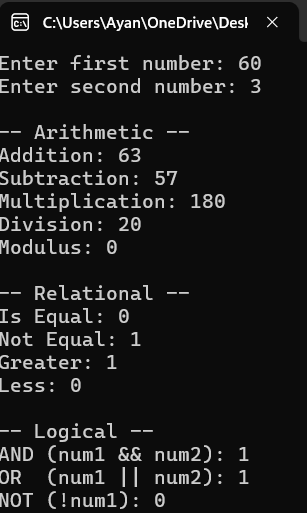
printf("AND (num1 && num2): %d\n", num1 && num2);

printf("OR (num1 || num2): %d\n", num1 || num2);

printf("NOT (!num1): %d\n", !num1);

return 0;

}



Q5. Explain decision-making statements in C (if, else, nested if-else, switch). Provide examples of each.

* **1]. If statement:-** Used to check one condition.

If the condition is **true**, the code inside runs.

* 2]. Else statement :- if condition is false than execute the code inside the else block.

Example :- #include<stdio.h>

int main()

{

int age;

printf("\nEnter your age = ");

scanf("%d",&age);

if(age>18)

{

printf("\nYou are eligible for voting");

}

else

{

printf("\nYou are not eligible for voting");

}

return 0;

}

* 3]. Nested if…else :- if inside another if. Used for checking **multiple levels** of conditions.

Example :-

#include<stdio.h>

int main()

{

int n1,n2,n3;

printf("\nEnter the value in n1 = ");

scanf("%d",&n1);

printf("\nEnter the value in n2 = ");

scanf("%d",&n2);

printf("\nEnter the value in n3 = ");

scanf("%d",&n3);

if(n1>n2)

{

if(n1>n3)

{

printf("\n%d is the biggest number",n1);

}

else

{

printf("\n%d is the biggest number",n3);

}

}

else

{

if(n2>n3)

{

printf("\n%d is the biggest number",n2);

}

else

{

printf("\n%d is the biggest number",n3);

}

}

return 0;

}

* 4]. Switch statement :- Used to **choose between many options.**

**Example :-**

#include <stdio.h>

int main() {

int day = 3;

switch (day) {

case 1:

printf("Monday\n");

break;

case 2:

printf("Tuesday\n");

break;

case 3:

printf("Wednesday\n");

break;

default:

printf("Invalid day\n");

}

return 0;

}

Q6. Write a C program to print numbers from 1 to 10 using all three types of loops (while, for, do-while).

* .

#include <stdio.h>

int main() {

int i;

// Using while loop

printf("Using while loop:\n");

i = 1;

while (i <= 10) {

printf("%d ", i);

i++;

}

printf("\n\nUsing for loop:\n");

// Using for loop

for (i = 1; i <= 10; i++) {

printf("%d ", i);

}

printf("\n\nUsing do-while loop:\n");

// Using do-while loop

i = 1;

do {

printf("%d ", i);

i++;

} while (i <= 10);

return 0;

}

Q7. Write a C program that uses the break statement to stop printing numbers when it reaches 5. Modify the program to skip printing the number 3 using the continue statement.

* 1. Break statement : -

#include <stdio.h>

int main() {

int i;

printf("Using break (stop at 5):\n");

for (i = 1; i <= 10; i++) {

if (i == 5) {

break; // Stop the loop when i is 5

}

printf("%d ", i);

}

return 0;

}

* 2. Continue statement :-

#include <stdio.h>

int main() {

int i;

printf("Using continue (skip 3):\n");

for (i = 1; i <= 5; i++) {

if (i == 3) {

continue; // Skip when i is 3

}

printf("%d ", i);

}

return 0;

}

Q8. Write a C program that calculates the factorial of a number using a function. Include function declaration, definition, and call.

* Code;

#include <stdio.h>

int factorial(int n);

int main()

{

int num, result;

printf("Enter a number: ");

scanf("%d", &num);

result = factorial(num);

printf("Factorial of %d is %d\n", num, result);

return 0;

}

int factorial(int n)

{

int fact = 1;

for (int i = 1; i <= n; i++)

{

fact \*= i; // fact = fact \* i

}

return fact;

}

Q9. Write a C program that stores 5 integers in a one-dimensional array and prints them. Extend this to handle a two-dimensional array (3x3 matrix) and calculate the sum of all elements.

* 1. 1d array :

#include <stdio.h>

int main() {

int i;

int numbers[5] = {10, 20, 30, 40, 50};

printf("1D Array Elements:\n");

for (i = 0; i < 5; i++)

{

printf("%d ", numbers[i]);

}

return 0;

}

* 2. 2d array sum :-

#include <stdio.h>

int main()

{

int matrix[3][3];

int i, j, sum = 0;

printf("Enter elements of 3x3 matrix:\n");

for (i = 0; i < 3; i++)

{

for (j = 0; j < 3; j++)

{

printf("Enter value at position [%d][%d]: ", i, j);

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

sum += matrix[i][j]; // Add each element to sum

}

}

printf("\nMatrix is:\n");

for (i = 0; i < 3; i++)

{

for (j = 0; j < 3; j++)

{

printf("%d ", matrix[i][j]);

}

printf("\n");

}

printf("\nSum of all elements = %d\n", sum);

return 0;

}

Q10. Write a C program to demonstrate pointer usage. Use a pointer to modify the value of a variable and print the result.

* ..

#include <stdio.h>

int main() {

int num = 10;

int \*ptr;

ptr = &num;

printf("Before change:\n");

printf("Value of num = %d\n", num);

\*ptr = 20;

printf("\nAfter change (using pointer):\n");

printf("Value of num = %d\n", num);

return 0;

}

Q11. Write a C program that takes two strings from the user and concatenates them using strcat(). Display the concatenated string and its length using strlen().

* ..

#include <stdio.h>

#include <string.h>

int main() {

char str1[100], str2[100];

printf("Enter first string: ");

gets(str1);

printf("Enter second string: ");

gets(str2);

strcat(str1, str2);

printf("\nConcatenated string: %s\n", str1);

printf("Length of concatenated string: %lu\n", strlen(str1));

return 0;

}