

Short Question :-

1) Differentiate between compiler and interpreter?

Ans A compiler translates the entire program into machine code at once before execution, making execution faster but error detection slower. An interpreter translates and executes the program line-by-line at runtime, making execution slower but error detection immediate.

2) Differentiate between object code and executable code?

Ans Object code is the intermediate machine code generated by a compiler after translating source code but before linking, often containing unresolved references. Executable code is the final machine code that can be directly executed by the operating system, typically created after linking all necessary object code files and libraries.

3, Write an algorithm to find the sum of two numbers?

Ans Algorithm to find the sum of two numbers

1. Start.

2. Input the first number (A).

3. Input the second number (B).

4. Calculate the sum: $\text{Sum} = A + B$.

5. Output the sum.

6. End.

4) Convert the given decimal number into its equivalent octal and hexadecimal number system? 84, 2.92

Ans

Here are the conversions of the given decimal numbers into their equivalent octal and hexadecimal representations:

1. Decimal Number: 84

• Octal (Base 8):

• $84 \div 8 = 10$ remainder 4

• $10 \div 8 = 1$ remainder 2

• $1 \div 8 = 0$ remainder 1

Octal Representation: 124

• Hexadecimal (Base 16):

• $84 \div 16 = 5$ remainder 4

• $5 \div 16 = 0$ remainder 5

Hexadecimal Representation: 54

2. Decimal Number: 92

• Octal (Base 8):

• $92 \div 8 = 11$ remainder 4

• $11 \div 8 = 1$ remainder 3

• $1 \div 8 = 0$ remainder 1

Octal representation: 134

• Hexadecimal (Base 16):

• $92 \div 16 = 5$ remainder 12 (12 is represented as C in hexadecimal)

• $5 \div 16 = 0$ remainder 5

Hexadecimal representation: 5C

Summary

• For 84:

• Octal: 124

• Hexadecimal: 54

• For 92:

• Octal: 134

• Hexadecimal: 5C

Long Question

1) Draw the block diagram of computer? Explain the each component of computer system in detail?

Ans Block diagram of computer system

* Input devices:

The input devices is used to enter data into a computer.

Eg:- Keyboard, Mouse, Touch screen and Scanner.

* CPU Central processing unit:

It is the main part of the computer.

It's main junction is to execute program store a main memory.

It consists of three functional unit
(ALU) Arithmetic and logical unit
(CU) control unit

(MMU) main memories unit

- Arithmetic and logical unit

It performs Arithmetic and logical operation on the data.

(Arithmetic means) : Addition, subtraction
(logical)

~~÷, ×, greater~~ division, multiplication, greater than, less than etc. and equal than

- Control unit :-

It controls the over all Activities of the component such as receiving (fetching the data) in storing the data, processing the data and sending the data in the computer.

- * Main memory unit :-

It has been classified into two types they are;

- * primary memory

- * volatile memory

- * non volatile memory

primary and volatile memory

It means that the information present in this types of memory is lost as soon as the power is switch off.

Primary memory is ~~again~~ classified into two types are.

1. RAM (Random Access memory)
2. Rom (Read only memory)

RAM classified into two types: these are

1. S-RAM (Static Rom)
2. D Rom (Dynamic Rom)

Rom again classified into three type these are

- P Rom (program Rom)
- E P Rom (Erasable Rom)
- E EP Rom (electrically erasable P Rom)

- * Rom is non volatile memory
- * It means the information is not least even through the power is switched off.

Secondary memory:

- * it is a non volatile memory

Ex: *

- * pen drive

- * magnetic tape.

- * optical disc etc.

2) Write an algorithm for finding the sum of all digits in a given number.

Ans: Algorithm to find the sum of all digits in a given number.

1. Start
2. Input the number (N)
3. Initialize sum to 0
4. While N is greater than 0:
 - (a) Get the last digit: $\text{Digit} = N \bmod 10$
 - (b) Add digit to sum: $\text{Sum} = \text{sum} + \text{Digit}$
 - (c) Remove the last digit from N: $N = N // 10$
5. Output the sum
6. End

3) Define flowchart symbol and its function.
Draw a flowchart to check whether a given number is prime or not.

Ans: Flowchart symbols and their functions
Flowchart use specific symbols to represent different types of actions or steps in a process. Here are some commonly used flowchart symbols and their functions:

1. Oval (Start/End): Represents the start and end points of the flowchart.
2. Rectangle (process): Indicates a process or operation, such as calculations or actions.

3. Diamond (Decision): Represents a decision point, where the flow can branch based on a yes/no question.

4. Parallelogram (Input/Output): Used for inputting data or displaying output.

5. Arrow (Flow line): Indicates the direction of flow in the process.

Flowchart to check if a Number is Prime:

1. Start

2. Input number

3. If $N \leq 1$, Print "Not prime"

4. Set is prime = 1

5. Loop: check divisibility of N by numbers from 2 to $\sqrt{N}$

6. If divisible, set is prime = 0

7. End Loop

8. If is prime = 1, print "prime", else print "Not prime"

9. End

This is a concise flow for determining if a number is prime.

- Q. What are the different types of operators that are included in C?
Give example for each.

A. In C operators are used to perform operations on variables and values.

Here are the main types of operators.

1. Arithmetic operators;

* $+$, $-$, $*$, $/$, $\%$

Ex: $a + b$, $a - b$, etc.

2. Relational Operators:

* $>$, $<$, $==$, $!=$

Ex: $a > b$

3. Logical operators:

* $==$, $!=$, $>$, $<$, $>=$, $<=$

Ex: $a > b$

4. Bitwise operators:

* $\&, |, \wedge, \sim, \ll, \gg$

Ex: $a \& b$

5. Assignment Operators

* $=, +=, -=, *=, /=$

Ex: $a += b$

6. Increment / Decrement Operators.

* $++, --$

Ex: $a++$

7. Conditional (Ternary) operators:

* $\text{Condition} ? \text{expr 1} : \text{expr 2}$

Ex: $a > b ? a : b$

8. Size of operator:

* Returns the size of a type

Ex: size of (int)

a. Pointer Operators

* (& Address-of), * (dereference)

Ex: `int *p = &a;`

* Each of these operators serves different purposes in programming.

ASSIGNMENT - 2

Short answers questions:

Classify the different types of decision-making statements.

Decision-making statements allow the program to make decisions and execute statements conditionally.

• Types include:

→ if statement: Executes a block of code if a ~~start~~ condition is true.

→ if-else statement: Execute one block if a condition is true, another if it's ~~is~~ false.

→ nested if statement: An if or else if inside another if.

→ else if ladder: Multiple conditions are checked one after the other.

→ switch statement: Executes one block of code from many based on a variable's value.

2) write and explain the ~~of~~ syntax of for loop:

→ The for loop is used for iteration, allowing code to be executed repeatedly.

syntax.

```
for (initialization; condition; update)
```

```
{
```

```
    // code block to be executed
```

```
}
```

- initialization executed once before the loop starts.
- condition checked before each iteration. The loop starts runs as long as this is true.
- updates executed after each iteration.

3) What is an array? How to declare and initialize 1-D, 2-D array with an example.

Ans An array is a collection of elements of the same data type stored at contiguous memory locations.

1-D Array:

```
int arr[5] = {1, 2, 3, 4, 5};
```

// Declaration and initialization.

2-D array:

```
int arr[2][3] = {{1, 2, 3}, {4, 5, 6}};
```

// Declaration and initialization.

4) Define string and list out its functions:

→ A string is a sequence of characters terminated by a null character $\backslash 0$.

→ String functions include:

→ strcpy(): Copies one string to another.

→ strcmp(): Compares two strings.

→ strcat(): Concatenates two strings.

→ strlen(): finds the length of a string.

→ strtok(): Concatenates a specified number of characters from one string to another.

5) define precedence and associativity write the rules for it.

→ precedence determines the order in which operators are evaluated in an expression.

→ Associativity defines the direction of execution (left-to-right or right-to-left) for operators with the same precedence.

• Rules:

→ Operator with higher precedence are evaluated first.

→ Operators of the same precedence level follow associativity rules.

→ Parentheses can be used to alter the normal precedence.

Long Answer questions:

1) Explain various branching statement in C with examples.

Ans Branching statement allow changing the flow of execution based on conditions. In C, the main branching statements are:

→ if statement:

```
if (condition)
```

```
{
```

```
// code to execute if condition is true.
```

```
}
```

→ if else statement:

```
if (condition)
```

```
{
```

```
// code if true
```

```
}
```

```
else
```

```
{
```

```
// code if false.
```

```
}
```

→ nested if statement:

allowing using an if statement inside another if

→ Switch Statement:

Evaluates a variable and branch to code

blocks based on case labels.

Ex:-

Switch (expression)

{

case variable : // code

case variable : // code

:

default : // code

}

2. List and explain loop control (or iteration) statements in C.

Ans Loop control statements are used to repeat a block of code as long as the given condition is true.

→ 1) For loop:

Runs a block of code for a set number of times.

Syntax :-

for (initialization; condition; update)

{

// code

}

do While Loop: Repeats code as long as the condition is true.

Syntax:

```
while (condition)
{
    // Code
}
```

do-while loop:

Executes code at least once and then repeats while the condition is true.

Syntax:

```
do {
```

```
    // Code
```

```
} while (condition);
```

3) Explain the following string handling functions with examples:

Ans String copy (strcpy()) :- Copies source string to destination.

Ex:-

```
char src[] = "Hello";
```

```
char dest[20];
```

```
strcpy (dest, src);
```

→ strcmp(string compare):
• compare two strings.

Ex:-

```
int result = strcmp("Hello", "world");
```

→ strcat(): Concatenates two strings.

Ex:-

```
char dest[20] = "Hello";  
strcat(dest, "world");
```

→ strlen(): (string length).
Returns length of the string.

Ex int len = strlen("Hello");

→ strncat(): Concatenates a limited
number of characters from
source to destination.

Ex:-

```
char dest[20] = "Hello";  
strncat(dest, "world", 3);
```

4. Write a calculator program in C language to do simple operations like addition, subtraction, multiplication, and division using switch case.

Ans =

```
#include <stdio.h>
#include <math.h>
void main ()
{
    int a, b, sub, mul, add, div, mod, opt;
    printf("menu:\n 1. Addition\n 2. subtraction\n 3. multiplication\n 4. Division\n 5. module\n");
    printf("Enter option from menu:\n");
    scanf("%d", &opt);
    printf("%d %d", a, b);
    switch(opt)
    {
        case 1: add = a + b;
            printf("\n sum is = %d", add);
            break;
        case 2: sub = a - b;
            printf("\n subtraction is = %d", sub);
            break;
        case 3: mul = a * b;
            printf("\n product is = %d", mul);
            break;
        case 4: div = a / b;
            printf("\n quotient is = %d", div);
            break;
        case 5: mod = a % b;
            printf("\n Perende is = %d", mod);
            break;
    }
    getch();
}
```

