Q2. PEEPHOLE OPTIMIZATION IN COMPILER DESIGN

It is a type of code optimization performed on a small part of code. It is performed on very small set of instructions in segment of code

The small set of instructions or small part of code on which peephole optimization is performed is known as peephole or window

It basically works on theory of replacement in which a part of code is replaced by shorter and faster code without change in output

Objectives of peephole organization - to improve performanc
- to reduce memory footprin
- to reduce code size

PEEPHOLE OPTIMIZATION TECHNIQUES

1. Redundant load & Store Elimination - In this techinque redundany is eliminated

```
    INITIAL CODE              OPTIMIZED CODE
    y=x+5;                    y=x+5;
     i=y;                     i=y;
     z=i;                     w=y * 3;
    w = z* 3;
```

2. CONSTANT FOLDING - the code that can be simplified by user itself is simplified

INITIAL CODE	OPTIMIZED CODE
x = 2 * 3;	x = 6;

3. STRENGTH REDUCTION - operator that consume higher execution time are replaced by operators ~~in~~ consuming less execution time.

INITIAL CODE	INITIAL CODE
y = x * 2;	y = x / 2;
OPTIMIZED CODE	OPTIMIZED CODE
y = x + x ;	y = x >> 1 ;

4. NULL SEQUENCES - useless operations are deleted

5. COMBINE OPERATIONS - several operations are replaced by a single equivalent operation.

Q3 BASIC BLOCK

It is set of ~~d~~ statements. The basic blocks do not have any in and out branches except entry and exit. It means the flow of controls enter at the beginning and will leave at the end without any hault. The set of instructions of basic block executes in sequence.

The new basic block always starts from the first instructions and keep adding instructions until a jump or label is met.

The algorithm for the construction of basic blocks is given below:

→ ALGORITHM

→ INPUT

→ OUTPUT

→ METHOD

FLOW GRAPH

It is a directed graph. After partitioning an intermediate code into blocks, the flow of control among basic blocks is represented by a flow graph. An edge can flow from one block x to another x block's first instruction immediatly follows the x block's last instruction.

The following ways will describe the edge:

→ There is a conditional or unconditional jump from the end X to the starting of Y.

→ Y immediatly follows X in original order of 3 address code, and X does not end in an unconditional jump.

The purpose of flow graph is to depict that how the program control is being parsed among the blocks. It contains the flow of control information for the set of basic block.

It is also useful in the loop optimization.