Peephole optimization :-

→ Peephole is the machine dependent optimization.

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

→ It is done to improve the performance of target 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 work on the theory of replacing in which a part of code is replaced by shorter & faster code without change in output.

Objectives of Peephole optimization

1. To improve performance
2. To reduce memory foot print
3. To reduce code size

Techniques of Peephole optimization

① Redundant instruction Elimination :-

→ Redundant Load & Store instructions can be eliminated

e.g consider the type of transformation instructions

unoptimized code	optimized code
MOV R0, a	
MOV a, R0	MOV R0, a

We can eliminate 2nd instruction since a is already in R_0

Especially the redundant loads & stores can be ... in this type of transformations

(2) Removal of eliminating Unreachable code :- We can eliminate the unreachable instructions for e.g. following is a piece of ccode.

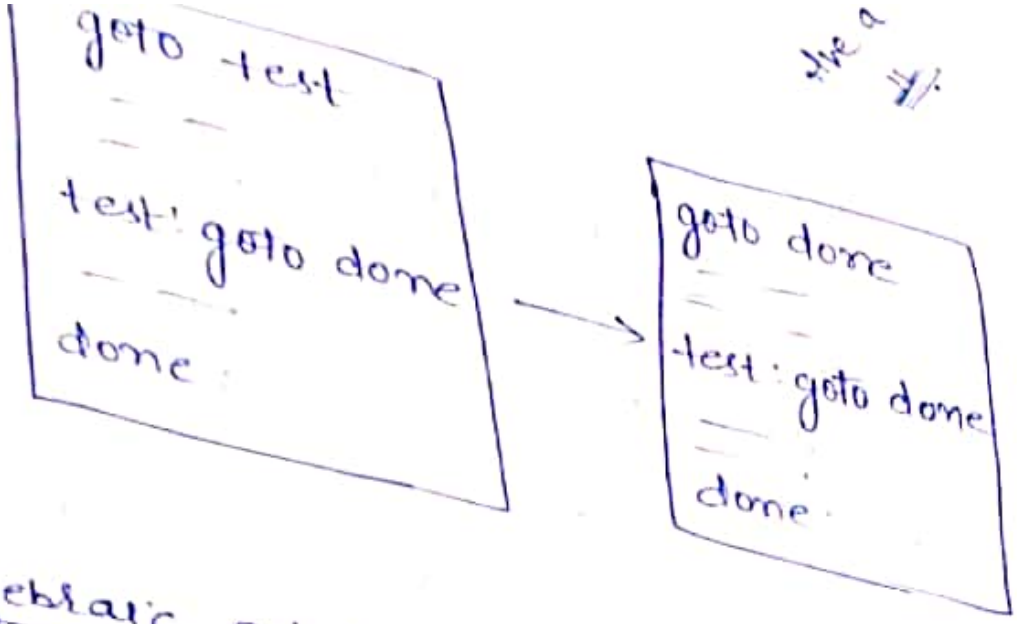
```
sum = 0
if (sum=1)
  printf("%d", sum);
```

Now this if statement will never get executed hence we can eliminate such a unreachable code similarly

eg:-
```
int fun(int a, int b)
  {
  c=a+b;
  return c;
  printf("%d", ch);  /* An unreachable code & hence eliminated */
  }
```

(3) Flow of control optimization

Using peephole optimization unnecessary jumps ~~on jumps~~ can be eliminated.



(4) Algebraic simplification:-

The statements such as

$x := x + 0$

or

$x := x \times 1$

can be eliminated by peephole optimization

- Above statements can be Eliminated because by executing those statements by result 'x' won't changes

(5) Reduction in strength:- The operator that consume higher execution time are replaced by the operator consuming less execution time.

Initial Code

```
y = x * 2;
```

optimized code

```
y = x + x;
```

6) Use of machine idioms :- The target instructions have equivalent machine instructions for performing some operations so we can replace these target instructions by equivalent machine instruction to improve efficiency

e.g- Auto increment instruction for x+1
eg inc x
Auto decrement instruction for x-1
dec x