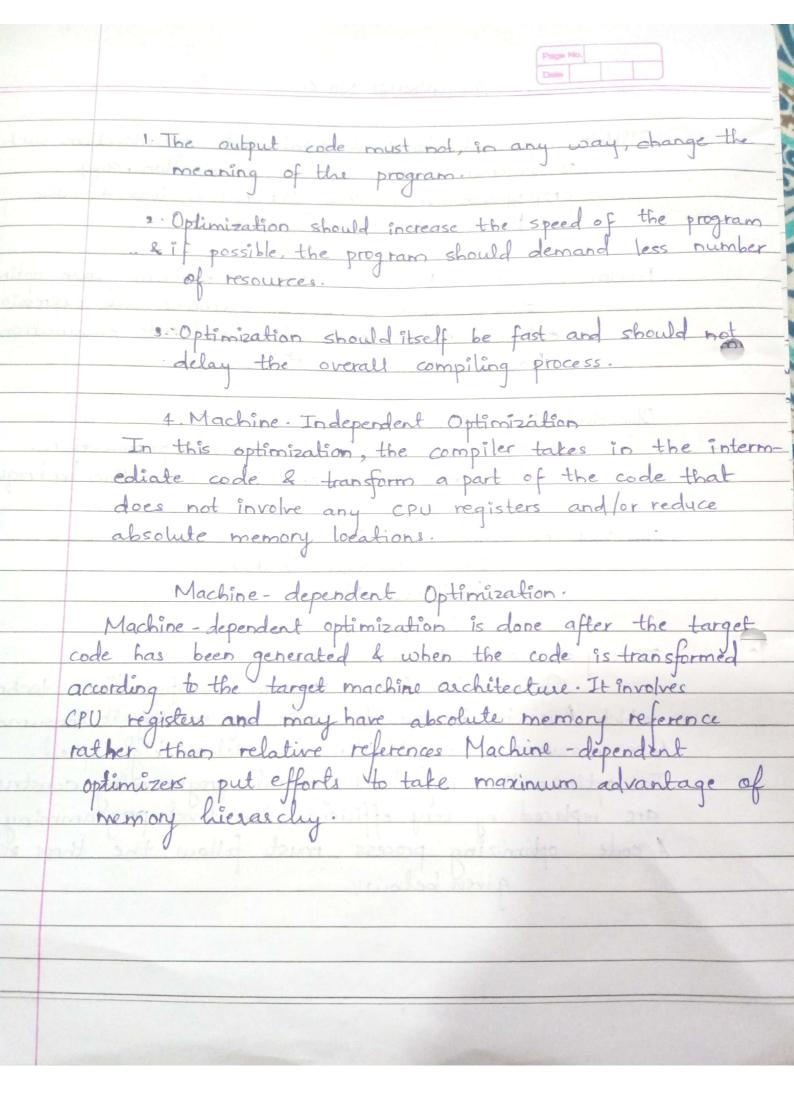
ASSIGNMENT NO:8. Title's Implement local & global code optimizations such as common such expression elimination, copy propagation, dead code elimination, loop & basic block optimization (optional) Problem Statement i WAP to implement different code optimi zation technique such as common sub expression elimination copy propagation, lead code elimination, loop 4 basic block optimization. Objective's To learn & understand · Different techniques of code optimization. · Implementation of code optimization techniques. s/w packages: C/C++ editors Linux OS/fedora/Ubunty. keyboard, Mouse. Theory's Code Optimization Optimization is a program transformation technique which tries to improve the code by making it consume less resources (i.e. CPU Memory) & deliver high speed. In optimization, high-level general programming constructs are replaced by very efficient low-level programming code A code optimizing process must follow the three rules given below:



	Page No.
	Code Optimization techniques.
	1. Common sub expression elimination.
	2. Copy propagation.
	3. Dual code elimination.
	4. loop & basic block optimization.
	ammon sub expression elimination.
	Common sub expression elimination is a compiler optimizate
8	on technique of finding redundant expression evaluation
	and replacing them with a single representational computational
	This saves the time overhead
	the expression for more than once.
	eg:
	In the followoode:
	q=b'+c+q';
	d=b+c+e;
	it may be worth transforming the code to:
242 21	trop = b = c;
	a=tmp+q;
10000	d=tmpte; d=tmpte; d=tmpte; d=dretrieving tmp is less than
	if the cost of storing and
	if the cost of storing and retrieving trop is less than) the cost of calculating b+c an extra time.
	Copy Propagation:
ambert.	Copy Propagation.
	Copy propagation is the process of replacing the occurrent of targets of direct assignments with their values. A
	of targets of direct assignment

	Page No.
	direct assignment is an instruction of the form x=y, which simply assigns the value of y to x.
	From the following code's y=x; Z=3+y; Copy propagation would yield: x=3+x;
	Algorithm's Generate the flow graph from list of instructions. do { 2) Perform reaching copy ma analysis.
	for each node in flow graph for each use if use is reached by the copy where it is the farget.
edir s	change the use to the source in the move statement tuple
s salt	White (changes) 4) Generate the list of instruction from modified flow program.

Page No. Dead Code Elimination Gode that is unreachable or that does not affect the - program (eg. dead stores) can be eliminated. the eg below, the value assigned to it is never used, & the dead store can be eliminated. The first assignment to global is dead, & 3rd assignment to global is unreacher ble both can be eliminated. int global; { int i; i=1; /* dead store */ global=1; /* dead store*/ global=3; /* unreachable */ Below is the code fragment after dead code elunination int global;
void f()
f global=2;
return;

