NAME: KRUNAL RANK

ROLL NO: U18CO081

CLASS: BTECH III, Computer Eng.

SEM: Semester 6

U18CO081

System Softwares

Tutorial 8

Ans1.

Let A be the absolute addresses - instructions or data addresses used in the instructions of program P.

A ≠ ϕ implies the program P assumes its instructions and data to occupy memory words with specific addresses

Such a program is called address sensitive program.

Relocation Example:-

	STATEMENTS	ADDRESS	MACHINE CODE
	ORIGIN 500		
	READ A	500	09 0 540
LOOP	. . .	501	
	.	.	.
A	DS 1	540	

If the program is loaded for execution in the memory address 900, its load time origin = 900

The load time address of loop would be 901
The Relocation factor = 900 - 500 = 400
Address of A = 940

inking Example:-

nsider an application program consisting of a set of programs
$S = \{P_i\}$

A program unit P_i interacts with another program P_j by using
ddresses of program P_j

o realize such interactions, P_j and P_i must contain public
finitions and external references defined in the following.

· ublic definition:- a symbol pub_symb defined in a program unit
hich may be referenced in other program units.

· xternal reference:- A reference to a symbol ext_symb which
s not defined in program unit.

· EXTERN and ENTRY statements:-

→ An ENTRY statement in program unit lists the public definitions
of a program UNIT

→ An EXTERN statement list the symbols to which external
eferences are made in the program

t self relocating program is a program which can perform
ts own relocation of address sensitive instructions.

It contains the following two provisions for this purpose:-

· A table of information concerning the address sensitive
instructions exists as a part of the program.

· Code to perform the relocation of address sensitive instructions
also exists as a part of the program. This is called relocating
logic

Ans 2. Static Linking:

- The linker loads all the modules of a program before the execution begins and produces a binary program that does not contain any unresolved external responses

- If statically linked programs use the same module from a library each program will get a private copy of the module

- If many programs that use the module are in execution at the same time, many copies of the module might be present in memory

Dynamic Linking

- It is performed during execution of a binary program
- The linker is invoked when an unresolved external reference is encountered during the execution of a program.
- The arrangement has several benefits concerning use, sharing and updating of library modules.
- If a module referenced by a program has already been linked to another program that is in execution, a copy of the module would exist in memory. The same copy of the module could be linked to his program as well, thus saving memory

Compile and Go Loaders-

- In this type of loader, the instruction is read line by line, its machine code is obtained and it is directly put in the main memory at some known address.
- That means the assembler runs in one part of memory and the assembled machine instructions and data directly put into their assigned memory locations.
- After the completion of loading ~~so~~ the assembler passes the control to the starting of instruction of loaded program.
- The loading scheme is called as 'assemble and go'.

Linker	Loader
• The main function of linker is to generate executible files.	• The main objective of loader is to load binary programs to main memory.
• The linker takes input of object code generated by compiler/assembler.	• The loader takes executible files as input
• It is the process of combining source program with external referenced modules into a single binary program.	• It is the process of loading executible codes to the main memory
• It is also responsible for arranging objects in program's address space.	• It is also responsible for adjusting references which are used within the program