

## PCC Assignment-2

- LAKSH SACHDEV (2022UCS1872)

### 1) Character set of Hindi language

Vowels - अ, आ, इ, ई, उ, ऊ, औ, ओ

Consonants ख ख ख ख (overall 33)

Numbers - ०, १, २, ३, ४, ५, ६, ७, ८

### 2) Different storage allocation strategy

#### ① static Allocation

- Memory allocated at compile time
- Once allocated memory can't be resized or freed until program ends.
- Suitable for fixed size data structures

#### ② stack Allocation

- Memory is managed using a stack
- Supports local variables in functions, using LIFO allocation.
- Memory is freed automatically upon function exit.

#### ③ heap Allocation

- Memory allocated at runtime
- Dynamic mem allocated using explicit allocation.
- Suitable for data structures that grow or shrink dynamically

#### ii) Automatic allocation

- Managed by language with garbage collection
- Allocate memory automatically when objects are created



#### Backpatching

→ Technique in compiler design used for filling in jump addresses when generating intermediate code for conditional statements & loops.

Helps managing forward references efficiently, especially

when address to jump is unknown until later in code generation process.

Example →

i) (acb) then  $t=1$  else  $t=0$

ii) if (acb) goto =<sup>y</sup>

2)  $t=0$

3) go to 5

4)  $t=1$

5)

#### 4) Types of Coercion

Coercions is automatic or implicit conversion of data types in programming. The main types of coercions are :-

① Implicit coercion → The computer automatically convert one

data type to another.

\* common in language that supports type promotion  
(converting int to float)

\* ensure operations proceed without explicit casts  
by the programmer.

② Explicit coercion (Casting) →

\* explicit conversion of data

from one type to another.

\* from int to int in explicit

③ Widening Coercion →

converts a type to a larger or more  
inclusive type (int to double)

\* minimizes data loss

④ Narrowing →

Converts a type to smaller or less inclusive  
type (double to int)

A may lead to data loss.

5) Syntax Directed Definitions to produce Tree-Address code for  
Booleans

SDD → provide rules for translating exp. in source code to  
intermediate representation like three address code

TAC for Boolean expression

- \* Use temporary variable to store intermediate results.
- \* Conditional branch for true and false evaluations

Ex

$\begin{array}{c} E \rightarrow E_1 \text{ or } M \\ E \cdot \text{true} = \text{merge } (E_1 \cdot \text{true}, M_1 \cdot \text{true}) \\ E \cdot \text{false} = E_2 \cdot \text{false} \\ L \rightarrow E_1 \text{ and } M_2 \end{array}$