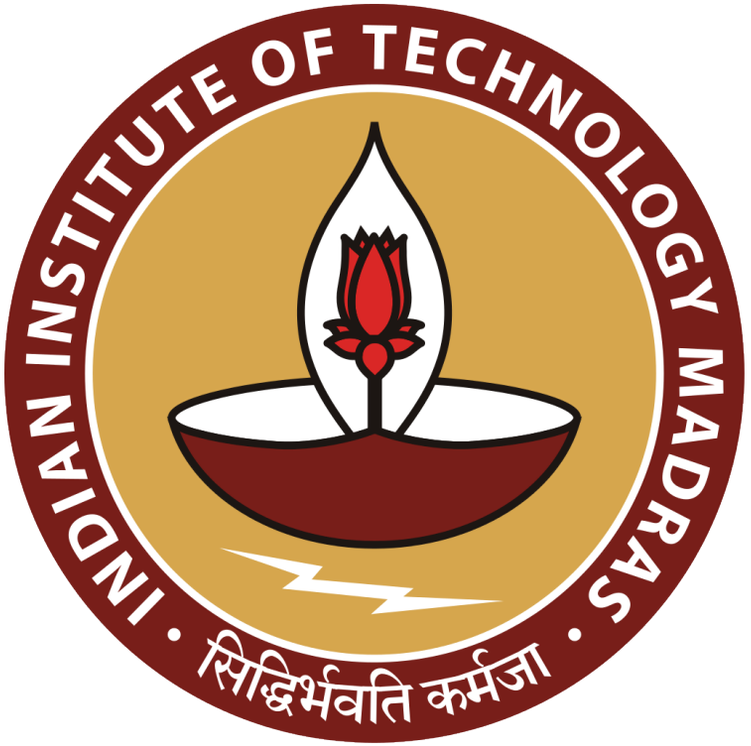


# IIT Madras

## ONLINE DEGREE

# Pseudocode: List example, top students

# Identifying top students

- Find students who are doing well in all subjects
  - Among the top 3 marks in each subject

# Identifying top students

- Find students who are doing well in all subjects
  - Among the top 3 marks in each subject
- Procedure for third highest mark in a subject

```
Procedure TopThreeMarks(Subj)
  max = 0, secondmax = 0, thirdmax = 0
  while (Table 1 has more rows) {
    Read the first row X in Table 1
    if (X.Subj > max) {
      thirdmax = secondmax
      secondmax = max
      max = X.Subj
    }
    if (max > X.Subj and X.Subj > secondmax) {
      thirdmax = secondmax
      secondmax = X.Subj
    }
    if (secondmax > X.Subj and X.Subj > thirdmax) {
      thirdmax = X.subj
    }
    Move X to Table 2
  }
  return(thirdmax)
End TopThreeMarks
```

# Identifying top students

- Find students who are doing well in all subjects
  - Among the top 3 marks in each subject
- Procedure for third highest mark in a subject
- Use lists
  - Construct a list of top students in each subject
  - Identify students who are present in all three lists

```
Procedure TopThreeMarks(Subj)
  max = 0, secondmax = 0, thirdmax = 0
  while (Table 1 has more rows) {
    Read the first row X in Table 1
    if (X.Subj > max) {
      thirdmax = secondmax
      secondmax = max
      max = X.Subj
    }
    if (max > X.Subj and X.Subj > secondmax) {
      thirdmax = secondmax
      secondmax = X.Subj
    }
    if (secondmax > X.Subj and X.Subj > thirdmax) {
      thirdmax = X.subj
    }
    Move X to Table 2
  }
  return(thirdmax)
End TopThreeMarks
```

# Constructing the lists

- Obtain cutoffs in each subject

```
cutoffMaths = TopThreeMarks(Mathematics)
cutoffPhys = TopThreeMarks(Physics)
cutoffChem = TopThreeMarks(Chemistry)
```

# Constructing the lists

- Obtain cutoffs in each subject
- Initialize lists for each subject

```
cutoffMaths = TopThreeMarks(Mathematics)
cutoffPhys = TopThreeMarks(Physics)
cutoffChem = TopThreeMarks(Chemistry)

mathsList = []
physList = []
chemList = []
```

# Constructing the lists

- Obtain cutoffs in each subject
- Initialize lists for each subject
- Scan each row
- For each subject, check if the marks are within the top three
- If so, append to the list for that subject

```
cutoffMaths = TopThreeMarks(Mathematics)
cutoffPhys = TopThreeMarks(Physics)
cutoffChem = TopThreeMarks(Chemistry)

mathsList = []
physList = []
chemList = []

while (Table 1 has more rows) {
   Read the first row X in Table 1
   if (X.Mathematics >= cutoffMaths) {
     mathsList = mathsList ++ [X.SeqNo]
   }
   if (X.Physics >= cutoffPhys) {
     physList = physList ++ [X.SeqNo]
   }
   if (X.Chemistry >= cutoffChem) {
     chemList = chemList ++ [X.SeqNo]
   }
   Move X to Table 2
}
```

# Find the overall toppers

- First find students who are toppers in Maths and Physics

```
mathsPhysList = []
foreach x in mathsList {
   foreach y in PhysList {
     if (x == y) {
       mathsPhysList = mathsPhysList ++ [x]
     }
   }
}
```

# Find the overall toppers

- First find students who are toppers in Maths and Physics

```
mathsPhysList = []
foreach x in mathsList {
   foreach y in PhysList {
     if (x == y) {
       mathsPhysList = mathsPhysList ++ [x]
     }
   }
}
```

- Then match these toppers with toppers in Chemistry

```
mathsPhysChemList = []
foreach x in mathsPhysList {
   foreach y in chemList {
     if (x == y) {
       mathsPhysChemList =
                    mathsPhysChemList ++ [x]
     }
   }
}
```

# Summary

- Lists are useful to collect items that share some property

# Summary

- Lists are useful to collect items that share some property
- Nested iteration can find common elements across two lists

# Summary

- Lists are useful to collect items that share some property
- Nested iteration can find common elements across two lists
- Can group lists to process more than two lists

# Summary

- Lists are useful to collect items that share some property
- Nested iteration can find common elements across two lists
- Can group lists to process more than two lists
  - Find common items across four lists, `list1`, `list2`, `list3`, `list4`

# Summary

- Lists are useful to collect items that share some property
- Nested iteration can find common elements across two lists
- Can group lists to process more than two lists
  - Find common items across four lists, `list1`, `list2`, `list3`, `list4`
  - Nested iteration on `list1`, `list2` constructs `list12` of common items in first two lists

# Summary

- Lists are useful to collect items that share some property
- Nested iteration can find common elements across two lists
- Can group lists to process more than two lists
  - Find common items across four lists, `list1`, `list2`, `list3`, `list4`
  - Nested iteration on `list1`, `list2` constructs `list12` of common items in first two lists
  - Nested iteration on `list3`, `list4` constructs `list34` of common items in last two lists

# Summary

- Lists are useful to collect items that share some property
- Nested iteration can find common elements across two lists
- Can group lists to process more than two lists
  - Find common items across four lists, `list1`, `list2`, `list3`, `list4`
  - Nested iteration on `list1`, `list2` constructs `list12` of common items in first two lists
  - Nested iteration on `list3`, `list4` constructs `list34` of common items in last two lists
  - Nested iteration on `list12`, `list34` finds common items across all four lists