

Advanced Programming

Arrays and Structs

Topics

- Arrays
 - Vector arrays
 - 2 dimensional arrays
 - Higher dimensional arrays
- Strings
 - Strings as character arrays
 - Processing strings
- Structs
- Unions

Arrays

- Arrays are a set of variables with
 - Same name
 - Same type
 - Located sequentially in the memory
- To access the elements of an array we use indexes

Vector Arrays

- Declaration

`type name[number of elements];`

- indexes always start with zero
- C does not check the range of indexes
- loop variables can be used as array indexes

Example

- Use an integer vector to store a sequence of numbers
 - Find the smallest number
 - Sort the array

2 Dimensional Arrays

- 2 Dimensional arrays or matrices are declared as:

```
type name[rows][columns];
```

- both row and column indexes start from zero

Example

- Use a 2D array to store an integer 5x5 matrix.
 - Read data into matrix.
 - Find the sum of its main diagonal elements.
 - Find transpose of the matrix
- Use two integer matrices and find their multiplication.

Higher Dimensional Arrays

- Declaration:

```
type name[dim1][dim2][dim3]....[dimn];
```

C arrays are row-major

Strings

- Strings are character arrays
- A string should be terminated with a null character (`'\0'`)
- Constant strings are given between “ ”

String Processing Functions

- (include string.h)
 - strlen
 - strcmp
 - strcat
 - strncpy
 - strncat
 - memset
 - strstr

Example

- Read the name and grades of 10 students (5 grade per student). Find the student with the highest average and print it

Structs

- Structs are user defined types where:
 - Has a name
 - May have fields with different types
 - Each field is referred to by its name
- struct name

```
{ type1 field1;
  tyep2 field2;
}
```

Example

- Create a table where each row has the name, surname, ID , and phone number of a student.
- Read data into table
- Read a phone number and find the corresponding student

Unions

- User defined types with multiple fields
- Fields are accessed by name
- The difference with structs is that in unions the fields overlap

Example

- The list of employees in a company contains:
 - Name
 - Surname
 - Gender
 - Responsibility
 - If manager then office number
 - If engineer then project code