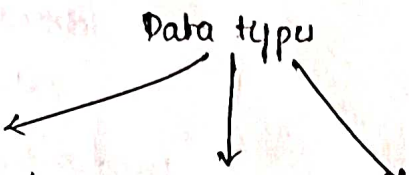
7. Arrays

* Data structures.



derived Types	fundamental types	user-defined types
- Arrays	- integral types	- structures
- function	- float	- union
- pointers	- character	- Enumerations

Including
- Linked lists
- Stacks
- Queues
- Trees

* one-dimensional array:
A list of items can be given one variable name using only one subscript and such a variable is called a one-dimensional array

x[0].

* Declaration of one dimensional arrays:
arrays must be declared before they are used so that the compiler can allocate space for them in memory. The general form of array declaration is

type variable-name [size];

Eg: float height[50];

* Initialization of one-dimensional arrays:
After an array is declared, its elements must be initialized. Otherwise, they will contain "garbage". An array can be initialized at either in two ways,

* At compile time

type array-name[size] = {list of values};

Eg: int number[3] = {0,0,0};

* At run time initialization.

An array can be explicity initialized at run time. This approach is useful appired for initializing large arrays.

1) Find two's complement of a binary number

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
  char a[16];
  int i,j,k,len;
  clrscr();
  printf("\nEnter a binary number\n");
  gets(a);
  len=strlen(a);
  for(k=0;a[k]!='\0';k++)
  {
    if(a[k]!='0' && a[k]!='1')
    {
      printf("\nIncorrect binary number format");
      getch();
      exit(0);
    }
  }
  for(i=len-1;a[i]!='1';i--)
  for(j=i-1;j>=0;j--)
  {
    if(a[j]=='1')
    a[j]='0';
    else
      a[j]='1';
  }
  printf("\n 2's compliment =%s",a);
  getch();
}
```

1) two NXN matrices
2) find transpose of a matrix

* Searching and Sorting

Sorting is the process of arranging elements in the list according to their values, in ascending or descending order. A sorted list is called an ordered list

- Bubble sort
- Selection sort
- Insertion sort

Includes shell sort, merge sort and quick sort

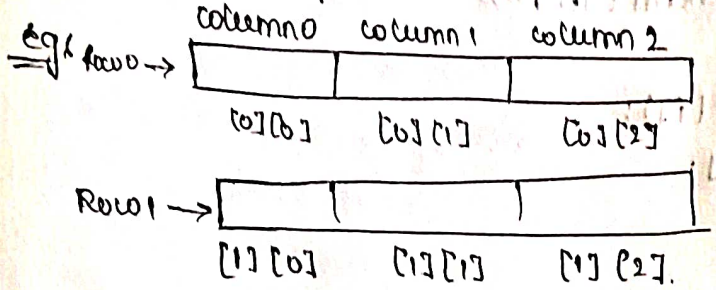
Searching is the process of finding the location of the specified element in a list. The specified element is often called the search key

- Sequential search
- Binary search.

* Two dimensional arrays

Two dimensional arrays are declared as follows

type array-name [row_size] [column_size];



Initializing

int table [2] [3] = { 0,0,0, 1,1,1 }
(Row 0, Row 1)

* Multi dimensional arrays

C allows arrays of three or more dimensions. The exact limit is determined by the compiler. The general form of a multidimensional array is

type array-name [s1] [s2] [s3] ... [sm];

Eg: int survey [3] [5] [12];

8) Read and display elements of an array

```
#include <stdio.h>
int main() {
    int arr[100], n, i;
    printf("Enter number of elements:");
    scanf("%d", &n);
    printf("Enter %d elements:\n", n);
    for(i=0; i<n; i++)
        scanf("%d", &arr[i]);
    printf("Array elements are:\n");
    for(i=0; i<n; i++)
        printf("%d", arr[i]);
    return 0;
}
```

3) Sum and average of array elements

```
#include <stdio.h>
int main() {
    int arr[100], n, i, sum=0;
    float avg;
    printf("Enter number of elements:");
    scanf("%d", &n);
    printf("Enter %d elements:\n", n);
    for(i=0; i<n; i++){
        scanf("%d", &arr[i]);
        sum = sum+arr[i];
    }
    avg = (float) sum/n;
    printf("sum = %d\n", sum);
    printf("Avg = %.2f\n", avg);
    return 0;
}
```

4. Largest and smallest elements in an array

```
#include <stdio.h>
int main(){
    int arr[100], n, i, max, min;
    printf("Enter number of elements:");
    scanf("%d", &n);
    printf("Enter %d elements:\n", n);
    for(i=0; i<n; i++)
        scanf("%d", &arr[i]);
    max = min = arr[0];
    for(i=1; i<n; i++){
        if(arr[i] > max)
            max = arr[i];
        if(arr[i] < min)
            min = arr[i];
    }
    printf("Largest = %d\n", max);
    printf("Smallest = %d\n", min);
    return 0;
}
```

supriya ✨

5) Reverse the elements of an array

```
#include<stdio.h>
int main(){
    int arr[100],n,i;
    printf("\n enter no. of elements\n");
    scanf("%d",&n);
    printf("\n enter %d elements:\n",n);
    for(i=0;i<n;i++)
        scanf("%d",&arr[i]);
    printf("\n Reversed array:\n");
    for(i=n-1;i>=0;i--)
        printf("%d",arr[i]);
    return 0;
}
```

```c
/* Fitting a Straight Line*/
#include <stdio.h>
int main()
{
        int n, i;
        float x[100], y[100];
        float sumx=0, sumy=0, sumxy=0, sumx2=0;
        float m, c;
        printf("Enter number of points: ");
        scanf("%d", &n);
        printf("Enter x and y values:\n");
        for(i=0; i<n; i++)
        {
                scanf("%f %f", &x[i], &y[i]);
                sumx += x[i];
                sumy += y[i];
                sumxy += x[i]*y[i];
                sumx2 += x[i]*x[i];
        }
        m = (n*sumxy - sumx*sumy) / (n*sumx2 - sumx*sumx);
        c = (sumy - m*sumx) / n;
        printf("Equation of line: y = %.2fx + %.2f\n", m, c);
        return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp7$ ./str
Enter number of points: 6
Enter x and y values:
2 3
5 6
3 9
1 2
4 5
3 9
Equation of line: y = 1.00x + 1.00
```

```c
/* Election Voting*/
#include <stdio.h>
int main()
{
        int votes[100], n, i, count[6]={0}, ballot;
        printf("Enter number of ballots: ");
        scanf("%d", &n);
        printf("Enter votes (1-5):\n");
        for(i=0; i<n; i++)
        {
                scanf("%d", &ballot);
                if(ballot>=1 && ballot<=5)
                        count[ballot]++;
                else
                        count[0]++;
         }
        for(i=1; i<=5; i++)
                printf("Candidate %d: %d votes\n", i, count[i]);
        printf("Spoilt ballots: %d\n", count[0]);
        return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp7$ ./ele
Enter number of ballots: 5
Enter votes (1-5):
1 2 3 4 5
Candidate 1: 1 votes
Candidate 2: 1 votes
Candidate 3: 1 votes
Candidate 4: 1 votes
Candidate 5: 1 votes
Spoilt ballots: 0
```

```c
/* Pascal's Triangle*/
#include <stdio.h>
int main()
{
        int n=10, i, j;
        int a[20][20] = {0};
        for(i=0; i<n; i++)
        {
                a[i][0] = a[i][i] = 1;
                for(j=1; j<i; j++)
                        a[i][j] = a[i-1][j-1] + a[i-1][j];
        }
        printf("Pascal's Triangle (10 rows):\n");
        for(i=0; i<n; i++)
        {
                for(j=0; j<=i; j++)
                        printf("%4d", a[i][j]);
                printf("\n");
        }
        return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp7$ ./tri
Pascal's Triangle (10 rows):
   1
   1   1
   1   2   1
   1   3   3   1
   1   4   6   4   1
   1   5  10  10   5   1
   1   6  15  20  15   6   1
   1   7  21  35  35  21   7   1
   1   8  28  56  70  56  28   8   1
   1   9  36  84 126 126  84  36   9   1
```

```c
/* Merge Two Sorted Arrays */
#include <stdio.h>
int main()
{
        int A[50], B[50], C[100];
        int n1, n2, i, j, k=0;
        printf("Enter size of array A: ");
        scanf("%d", &n1);
        printf("Enter elements of A (sorted): ");
        for(i=0; i<n1; i++)
                scanf("%d", &A[i]);
        printf("Enter size of array B: ");
        scanf("%d", &n2);
        printf("Enter elements of B (sorted): ");
        for(i=0; i<n2; i++)
                scanf("%d", &B[i]);
        i=0; j=0;
        while(i<n1 && j<n2)
        {
                if(A[i] < B[j])
                        C[k++] = A[i++];
                else
                        C[k++] = B[j++];
        }
        while(i<n1)
                C[k++] = A[i++];
        while(j<n2)
                C[k++] = B[j++];
        printf("Merged Sorted Array: ");
        for(i=0; i<k; i++)
                printf("%d ", C[i]);
        return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp7$ ./merge
Enter size of array A: 3
Enter elements of A (sorted): 1 2 3
Enter size of array B: 5
Enter elements of B (sorted): 56 78 89 90 91
Merged Sorted Array: 1 2 3 56 78 89 90 91 supriya@ubuntu:~/Desktop/
```

```c
/* Matrix Multiplication*/
#include <stdio.h>
int main()
{
        int n, i, j, k;
        int A[10][10], B[10][10], C[10][10];
        printf("Enter size of matrix (n x n): ");
        scanf("%d", &n);
        printf("Enter elements of A:\n");
        for(i=0;i<n;i++)
                for(j=0;j<n;j++)
                        scanf("%d",&A[i][j]);
        printf("Enter elements of B:\n");
        for(i=0;i<n;i++)
                for(j=0;j<n;j++)
                        scanf("%d",&B[i][j]);
        for(i=0;i<n;i++)
                for(j=0;j<n;j++)
                {
                        C[i][j]=0;
                        for(k=0;k<n;k++)
                                C[i][j]+=A[i][k]*B[k][j];
                }
        printf("Product Matrix:\n");
        for(i=0;i<n;i++)
        {
                for(j=0;j<n;j++)
                        printf("%4d",C[i][j]);
                printf("\n");
        }
        return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp7$ ./mul
Enter size of matrix (n x n): 2 3
Enter elements of A:
1 2 3
Enter elements of B:
4 5 6
7 8 9
Product Matrix:
  18  22
  26  31
```

```c
/* Special 5x5 Matrix*/
#include <stdio.h>
int main()
{
        int a[5][5], i, j;
        for(i=0;i<5;i++)
        {
                for(j=0;j<5;j++)
                {
                        if(i<j)
                                a[i][j]=1;
                        else if(i>j)
                                a[i][j]=-1;
                        else a[i][j]=0;
                }
        }
        printf("Matrix:\n");
        for(i=0;i<5;i++)
        {
                for(j=0;j<5;j++)
                        printf("%3d", a[i][j]);
                printf("\n");
        }
        return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp7$ ./mat
Matrix:
  0  1  1  1  1
 -1  0  1  1  1
 -1 -1  0  1  1
 -1 -1 -1  0  1
```

```c
/* Selection Sort*/
#include <stdio.h>
int main()
{
        int a[50], n, i, j, max, temp;
        printf("Enter size: ");
        scanf("%d",&n);
        printf("Enter elements:\n");
        for(i=0;i<n;i++)
                scanf("%d",&a[i]);
        for(i=n-1;i>0;i--)
        {
                max=0;
                for(j=1;j<=i;j++)
                        if(a[j]>a[max])
                                max=j;
                temp=a[max];
                a[max]=a[i];
                a[i]=temp;
        }
        printf("Sorted list:\n");
        for(i=0;i<n;i++)
                printf("%d ",a[i]);
        return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp7$ ./sort
Enter size: 6
Enter elements:
89 45 2 1 56 90
Sorted list:
1 2 45 56 89 90 supriya@ubuntu:~/Desktop/c/chp7$
```

```c
/* Binary Search*/
#include <stdio.h>
int main()
{
        int a[50], n, i, key, low, high, mid, found=0;
        printf("Enter size: ");
        scanf("%d",&n);
        printf("Enter sorted elements:\n");
        for(i=0;i<n;i++)
                scanf("%d",&a[i]);
        printf("Enter key: ");
        scanf("%d",&key);
        low=0; high=n-1;
        while(low<=high)
        {
                mid=(low+high)/2;
                if(a[mid]==key)
                {
                        found=1;
                        break;
                }
                else if(key<a[mid])
                        high=mid-1;
                else
                        low=mid+1;
        }
        if(found)
                printf("Key found at position %d\n", mid+1);
        else
                printf("Key not found\n");
        return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp7$ ./search
Enter size: 3
Enter sorted elements:
1 2 3
Enter key: 2
Key found at position 2
```

```
supriya@ubuntu:~/Desktop/c/chp7$ cat len.c
/*Length of a String*/
#include <stdio.h>
#include <string.h>
int main() {
    char str[100];
    printf("Enter a string: ");
    fgets(str, sizeof(str), stdin);
    str[strcspn(str, "\n")] = '\0';
    printf("Length of string is: %lu\n", strlen(str));
    return 0;
}

supriya@ubuntu:~/Desktop/c/chp7$ ./len
Enter a string: supriya
Length of string is: 7
```

```c
/* Transpose of a Matrix*/
#include <stdio.h>
int main()
{
        int m, n, i, j, A[10][10], T[10][10];
        printf("Enter rows and columns: ");
        scanf("%d%d", &m, &n);
        printf("Enter matrix:\n");
        for(i=0;i<m;i++)
                for(j=0;j<n;j++)
                        scanf("%d",&A[i][j]);
        for(i=0;i<m;i++)
                for(j=0;j<n;j++)
                        T[j][i]=A[i][j];
        printf("Transpose:\n");
        for(i=0;i<n;i++)
        {
                for(j=0;j<m;j++)
                        printf("%4d",T[i][j]);
                printf("\n");
        }
        return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp7$ ./trans
Enter rows and columns: 2 3
Enter matrix:
1 2 3
4 56
7
Transpose:
   1   4
   2  56
   3   7
```

```c
/* Reverse the elements of an array*/
#include <stdio.h>
int main() {
    int arr[100], n, i;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++)
        scanf("%d", &arr[i]);
    printf("Reversed array:\n");
    for(i = n-1; i >= 0; i--)
        printf("%d ", arr[i]);
    return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp7$ ./reverse
Enter number of elements: 4
Enter 4 elements:
5 6 9 0
Reversed array:
0 9 6 5 supriya@ubuntu:~/Desktop/c/chp7$
```

```c
/*Copy elements from one array to another*/
#include <stdio.h>
int main() {
    int arr1[100], arr2[100], n, i;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++)
        scanf("%d", &arr1[i]);
    for(i = 0; i < n; i++)
        arr2[i] = arr1[i];
    printf("Copied array elements:\n");
    for(i = 0; i < n; i++)
        printf("%d ", arr2[i]);
    return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp7$ ./copy
Enter number of elements: 6
Enter 6 elements:
1 2 3 4 5 6
Copied array elements:
1 2 3 4 5 6 supriya@ubuntu:~/Desktop/c/chp7$
```

```
supriya@ubuntu:~/Desktop/c/chp7$ cat even.c
/*Count even and odd elements in an array*/
#include <stdio.h>
int main() {
    int arr[100], n, i, even = 0, odd = 0;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
        if(arr[i] % 2 == 0)
            even++;
        else
            odd++;
    }
    printf("Even elements = %d\n", even);
    printf("Odd elements = %d\n", odd);
    return 0;
}

supriya@ubuntu:~/Desktop/c/chp7$ ./even
Enter number of elements: 6
Enter 6 elements:
12 3 4 5 1 6
Even elements = 3
Odd elements = 3
supriya@ubuntu:~/Desktop/c/chp7$
```

```c
#include <stdio.h>
int main() {
    int arr[100], n, i, j;
    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    printf("Duplicate elements are:\n");
    for(i = 0; i < n; i++) {
        for(j = i+1; j < n; j++) {
            if(arr[i] == arr[j]) {
                printf("%d\n", arr[i]);
                break; // Avoid printing same duplicate multiple times
            }
        }
    }

    return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp7$ ./dup
Enter number of elements: 6
Enter 6 elements:
78 1 2 3 78 2
Duplicate elements are:
78
```

```c
/*Delete an element from an array at a given position*/
#include <stdio.h>
int main() {
    int arr[100], n, pos, i;

    printf("Enter number of elements: ");
    scanf("%d", &n);
    printf("Enter %d elements:\n", n);
    for(i = 0; i < n; i++)
        scanf("%d", &arr[i]);

    printf("Enter position to delete (1-%d): ", n);
    scanf("%d", &pos);

    if(pos < 1 || pos > n) {
        printf("Invalid position!\n");
    } else {
        for(i = pos-1; i < n-1; i++)
            arr[i] = arr[i+1];
        n--;
        printf("Array after deletion:\n");
        for(i = 0; i < n; i++)
            printf("%d ", arr[i]);
    }

    return 0;
}
```

```
supriya@ubuntu:~/Desktop/c/chp7$ ./del
Enter number of elements: 4
Enter 4 elements:
7 8 9 0
Enter position to delete (1-4): 4
Array after deletion:
7 8 9 supriya@ubuntu:~/Desktop/c/chp7$
```