CS19001: Programming and Data Structures Laboratory

DRC, SD, SB
CSE, IIT Kharagpur

February 5, 2025

Arrays

Arrays are our first example of *structured data*.

Declaration

```
int A[400];
float B[123];
```

Each of the elements A[0],A[1],...,A[399] is a variable of type int

Declaration and Initialization

```
int A[5] = { 51, 29, 0, -34, 67 };
char C[4] = { 'g', 'o', 'd', '\0' };
char C[4] =  "god";

printf("%d\n",A[5]); /*prints a garbage value*/
```

Nested loops

One or more loops can be nested inside another loop.

Sorting

Suppose you have an array A of n elements (say, integers). They are stored in the array locations

`A[0], A[1], ..., A[n-1]`

We want to rearrange these integers in such a way that after the rearrangement we have

`A[0] <= A[1] <= A[2] <= ..... <= A[n-1]`

The resultant array is **sorted**. There are many such sorting methods. One is bubble sort.

Bubble sort

Code

```
for (i=n-2; i>=0; --i)
{
  for (j=0; j<=i; ++j)
  {
    if (A[j] > A[j+1])
    {
      t = A[j];
      A[j] = A[j+1];
      A[j+1] = t;
    }
  }
}
```

A[4]=4,3,2,1

i,j: A $\rightarrow A'$

2,0: 4,3,2,1 $\rightarrow$ 3,4,2,1

2,1: 3,4,2,1 $\rightarrow$ 3,2,4,1

2,2: 3,2,4,1 $\rightarrow$ 3,2,1,4

bubble till position i=4-2=2.

1,0: 3,2,1,4 $\rightarrow$ 2,3,1,4

1,1: 2,3,1,4 $\rightarrow$ 2,1,3,4

bubble till position i=1

0,0: 2,1,3,4 $\rightarrow$ 1,2,3,4

bubble till position i=0

Random Integer Generation

Write a C program to generate random numbers between 0 and N.

```c
#include <stdio.h>
#include <stdlib.h>
int main()
{
    // Set a constant 'N' to 1000
    int N = 1000;
    // Loop through 10 times
    for (int i = 0; i < 10; i++) {
        // Generate a random number between 0 and N using
        // the rand() function
        int value = rand() % (N + 1);
        // Print the generated random value
        printf("%d ", value);
    }
    return 0;
}
```

Programming Assignments

Complete and submit during lab

Assignment 1

Read an array of 10 integral numbers. Check if there exists a consecutive sequence of 3 numbers that are in ascending order. If there exists such a sequence, output the first such sequence, else output "None".

For example, for the input array [3, 5, 2, 3, 4, 6, 7, 8], it should print [2, 3, 4]

Assignment 2

Read a set of 10 positive integral numbers and store them in an array. Print the array.

Segregate the even and odd numbers without using any additional arrays (that is, in your final array, the even numbers must appear first, followed by the odd numbers). Print the final array.

Note that: the even and odd numbers should appear in the output array in the same order among themselves as they are in the input array.
Do not use additional arrays.

For example, for an array having the numbers [34,35,9,16,19,21,22,24,7,79], the final contents of the array should be [34,16,22,24,35,9,19,21,7,79]

Assignment 3

Write a program that does the following:

Generate **two** random arrays of a specified size (n) with random integers between 1 and 100, and print them. The size n should be taken as input. *Hint:* use the `rand()` function defined in `stdlib.h`.

Sort the two arrays using the bubble sort algorithm described above. Print both the sorted arrays.

Merge the two sorted arrays into a third array, such that the third array is also sorted, and contains all the numbers in the first two arrays (along with their repetitions if any). Print the merged array.

Note that: The merge process itself should be such that the final output array is sorted. You should not just copy all elements in a big array and then sort the array.

Bonus Assignment [Merry-Go-Round]

You have to read an n element array, shift circularly (right or left) the array to m positions and print the array. For this, you have to implement the following steps:

1. Display the following to the user:
   ```
   List of Activities:
   0.Exit, 1.Enter Array, 2.Print Array,
   3.Right-Shift Array, 4.Left-Shift Array
   Enter Your Choice:
   ```
2. Take from user a choice (0, 1, 2, 3, 4).
3. If the user enter 0 as choice, then terminate the program.
4. For Choice 1, (a) user inputs the number of array elements, (b) user inputs all the elements of the array, and (c) you store the elements in the array.
   For Choice 2, print all the elements of the array.
   For Choice 3 and similarly Choice 4, (a) user inputs the number of positions to shift right (for 3) or left (for 4), and (b) you shift the elements in the array circularly. However, you may not print the shifted array here (since you already have a print option 2).
5. *Repeat* Step-1 (displaying list of activities).

Thank You