Tutorial

Programming & Data Structure: CS 11001

Section - 4/D

Department of Computer Science and Engineering

I.I.T. Kharagpur

Spring Semester: 2013 - 2014 (06.02.2014)

Download

Download the file tut060214.pdf from Programming & Data Structures ... of

http://cse.iitkgp.ac.in/~goutam

View the file using the command acroread & or xpdf &

I/O in 1-D Array

```
#include <stdio.h>
#define MAX 100
int main() // 1DarrayRW.c
{
    int n, data[MAX], i;

    printf("Enter data count: ");
    scanf("%d", &n);
    printf("Enter %d data:\n", n);
```

Read and Write Data: 1-D Array

```
    for(i=0; i<n; ++i)
                scanf("%d", &data[i]);
    printf("Input is:\n");
    for(i=0; i<n; ++i)
                printf("%d ", data[i]);
    putchar('\n');

    return 0;
}
```

Largest Data: 1-D Array

```
#include <stdio.h>
#define MAX 100
int main() // lrg1Darray.c
{
    int n=0, data[MAX], max, i;

    printf("Enter data, terminate by Ctrl-D:\n");
    while(scanf("%d", &data[n]) != EOF) n++;
    max = data[0];
    for(i=1; i<n; ++i)
        if(data[i] > max) max = data[i];
    printf("Largest: %d\n", max);
    return 0;
}
```

Largest Data Function: 1-D Array

```
int largest(int x[], int m){ // lrg1DarrayF.c
    int i, max;

    max = x[0];
    for(i=1; i<m; ++i)
        if(x[i] > max) max = x[i];
    return max;
}
```

Largest Data Function: 1-D Array

```
int largest(int x[], int m){ // lrg1DarrayF.c
    int i, max;

    max = x[0];
    for(i=1; i<m; ++i)
        if(x[i] > max) max = x[i];
    return max;
}
```

Tutorial V.1

Write a C program that reads $n(\geq 2)$ data in a 1-D array, finds the second largest element from the array (not while reading), and prints it.

Tutorial V.2

Write a function
`int secLargest(int x[], int n)` that takes an array of $n(\geq 2)$ elements and returns the second largest element.

Tutorial V.3

Write a C program that reads n data in a 1-D array. After that it reads the $(n+1)^{th}$ data and insert it at the 0^{th} location.

Tutorial V.4

Solve the problem of IV.3 by writing a function `void insAt0th(int x[], int n, int data)` which takes an array containing n elements and a data as parameters, and inserts *data* in the 0^{th} location of the array.

ASCII Code

Character data e.g. `'A'`, `'a'`, `'$'`, `'3'` are represented as 8-bit ASCII codes in C language. Following are a few codes:

Characters	ASCII Codes (decimal)
'0', '1', ···, '9'	48, 49, ···, 57
'A', 'B', ···, 'Z'	65, 66, ···, 90
'a', 'b', ···, 'z'	97, 98, ···, 122

Character I/O

A character can be read using `scanf()` and is printed using `printf()` functions. The format specification is `%c`. We can also use `getchar()` and `putchar()` to read a character and print a character.

```
#include <stdio.h>
int main(){ // getput.c
    char c ;

    c = getchar() ;
    putchar(c) ;
    putchar('\n') ;
    return 0 ;
}
```

Loop Termination

A character read by `getchar()` can be used to control a loop. As an example a loop can be terminated after reading the end of the line `'\n'` (newline) or at the end-of-file `EOF (Ctrl-D)`, or any particular character for that matter.

```
#include <stdio.h>
int main(){ // termination.c
    char c ;

    while((c = getchar()) != '\n')
                        putchar(c) ;
    putchar('\n') ;
    return 0 ;
}
```

ASCII to Number

A digit *d* is read as a character (ASCII code) from the *keyboard*. Its decimal *face-value* can be extracted by subtracting the ASCII code of character '0' from it.

```
#include <stdio.h>
int main(){ // asciiToNum.c
    char c ;
    c = getchar() ;
    printf("val(%c) = %d\n", c, c-'0');
    return 0 ;
}
```

Tutorial V.5

Write a C program that reads a sequence of decimal digits as characters (not more than 9 digits) and converts it to its integer value and prints it.
Input: 1234 - read as characters '1', '2', '3', '4'
Output: 1234

Tutorial V.6

Use the code of previous problem to write the function `int getAtoI()` that reads a sequence of digits as characters, computes the number and returns the value.
Input: `246` (read character by character)
Output: `246`

Tutorial V.7

Write a C program that will read a sequence of characters up to the end-of-file (EOF). Rotate every English language alphabet by three characters ('a' → 'd', 'b' → 'e', ···, 'x' → 'a', 'y' →, 'b', 'z' → 'c', 'A' → 'D', 'B' → 'E', ···, 'X' → 'A', 'Y' →, 'B', 'Z' → 'C'. All other characters remain unchanged.

Input: `I spent 1000/- rupees in SF.`

Output: `L vshqw 1000/- uxshhv lq VI.`