

C Programming Basics

Ritu Arora

Texas Advanced Computing Center

June 28th, 2012

Email: rauta@tacc.utexas.edu



Function to allocate memory for a 2-D array of type double, Usage in HW-7

```
double** allocArrayDouble(double **a, int M, int N){
    int i;
    /* allocate storage for an array of pointers */
    a = malloc(M * sizeof(double *));

    /* for each pointer, allocate storage for an array
       of double */
    for (i = 0; i < M; i++) {
        a[i] = malloc(N * sizeof(double));
    }

    return a;
}
```

Homework 7 Solution (1)

```
1. #include <stdio.h>
```

```
2. #include <stdlib.h>
```

```
3. void printMatrix(double **arrayA, int M, int N);
```

```
4. void matmul(double **matA, double **matB);
```

```
5. double** allocArrayDouble(double **a, int M, int N);
```

```
6. int main() {
```

```
7.     int Rows, Cols;
```

```
8.     int i, j, count;
```

```
9.     double **myMatrixA, **myMatrixB;
```

```
10.    Rows = 2;
```

```
11.    Cols = 2;
```

Homework 7 Solution (2)

```
12. myMatrixA = allocArrayDouble(myMatrixA, Rows, Cols);
13. myMatrixB = allocArrayDouble(myMatrixB, Rows, Cols);
14. count = 1;
15. for(i=0; i<Rows; i++){
16.     for (j=0; j<Cols; j++){
17.         myMatrixA[i][j] = count;
18.         myMatrixB[i][j] = count;
19.         count++;
20.     }
21. }
22. matmul(myMatrixA, myMatrixB);
23. free(myMatrixA);
24. free(myMatrixB);
25. return 0;
26. }
```

Homework 7 Solution (3)

```
27. void matmul(double **matA, double **matB) {
28.     int i,j,k,sum;
29.     int Rows = 2;
30.     int Cols = 2;
31.     double **myMatrixC =
                allocArrayDouble(myMatrixC,Rows,Cols);
27.     //perform matrix multiply
28.     for (i = 0; i < Rows; i++){
29.         for (j = 0; j < Cols; j++){
30.             sum = 0;
31.             for (k = 0; k < Cols; k++){
32.                 sum += matA[i][k] * matB[k][j];
33.             }
34.             myMatrixC[i][j] = sum;
35.         }
36.     }
```

Homework 7 Solution (4)

```
37. printMatrix(myMatrixC, Rows, Cols);
38. free(myMatrixC);
39. }
40. void printMatrix(double **arrayA, int M, int N){
41.     int i,j;
42.     printf("Matrix C is:\n");

43.     for (i = 0; i < M; i++){
44.         for (j = 0; j < N; j++){
45.             printf(" %lf ",arrayA[i][j]);
46.         }
47.         printf("\n");
48.     }
49.     printf("\n");
50. }
```

Homework 7 Solution (5)

```
51. double** allocArrayDouble(double **a, int M, int N) {
52.     int i;
53.     a = malloc(M* sizeof(double *));
54.     for (i = 0; i < M; i++) {
55.         a[i] = malloc(N * sizeof(double));
56.     }
57.     return a;
58. }
```

Overview of the Course

- Writing a Basic C Program
- Understanding Errors
- Comments Keywords, Identifiers, Variables
- Operators
- Standard Input and Output (Basic)
- Control Structures
- Standard Input and Output
- Arrays, Structures
- Functions in C
- Pointers
- Working with Files

All the concepts will be accompanied with examples.

Including Library File for Maths: mathExample.c

```
#include <stdio.h>
#include <math.h>
int main() {
    double myNum = 2.2;
    int times = 8 ;
    printf("Square root of %lf is: %lf\n",myNum, sqrt(myNum) );
    return 0;
}
```

Output:

```
Square root of 2.200000 is: 1.483240
```

Math Library on Ranger

(MinGW users can ignore this)

- Note that `math.h` is a header-file that just includes the declarations of the math functions – recall function prototypes
- The compiled definitions are in the math library
- Link your program to the math library by adding `-lm` to the gcc command

```
login3$ gcc -o mathExample mathExample.c
/tmp/ccOdwtYH.o: In function `main':
mathExample.c:(.text+0x2b): undefined reference to `sqrt'
mathExample.c:(.text+0x74): undefined reference to `pow'
collect2: ld returned 1 exit status
login3$ gcc -o mathExample mathExample.c -lm
login3$
```

User-Defined Header Files

- Useful in multi-module, multi-person software development effort
- Save the following code in a file named `head.h` and don't compile/run it

```
/* This is my little header file named head.h */  
#define HAPPY 100  
#define SPIT printf  
#define POOL {  
#define PEEL }
```

User-Defined Header Files

- This is how the file `head.h` can be included in any program, here `headTest.c`

```
#include <stdio.h>
#include "head.h" ←- Notice the quotes around file name
int main()
POOL
SPIT("This guy is happy: %d percent\n", HAPPY);
return(0);
PEEL
```

Output:

```
This guy is happy: 100 percent
```

File I/O

- File pointer is required for accessing files to read, write or append

```
FILE *fp;
```

- **fopen** function is used to open a file and it returns a file pointer

```
FILE *fopen(const char *filename, const char *mode);
```

- The modes in which a file can be opened

r - open **for reading**

w - open **for writing** (file need not exist)

a - open **for appending** (file need not exist)

r+ - open **for reading and writing**, start at beginning

w+ - open **for reading and writing** (overwrite file)

a+ - open **for reading and writing** (append **if** file exists)

- To close a file

```
int fclose(FILE *a_file);
```

P.S. details at <http://www.cplusplus.com/reference/clibrary/cstdio/fopen/>

File I/O: fileExample.c

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int i, myInt;
    FILE *ifp;
    char *mode = "r";
    ifp = fopen("in.txt", mode);
    if (ifp == NULL) {
        fprintf(stderr, "Can't open input file in.txt!\n");
        exit(1);
    }else{
        for (i=0; i<10; i++){
            fscanf(ifp, "%d", &myInt); ←-- fscanf is used for reading file
            printf("%d\n", myInt);      contents
        }
    }
    fclose(ifp);
    return 0;
}
```

Write to a File: writeToFile.c

```
#include <stdio.h>
```

```
int main() {
```

```
FILE *fp;
```

```
fp = fopen("in2.txt", "a+");
```

```
fprintf(fp, "\n%d", 7000);
```

```
fclose(fp);
```

```
return 0;
```

```
}
```

Opening the file in
append mode

fprintf is used for
writing data to a file



Homework 8

- Redo the matrix multiplication problem assigned in the previous homeworks using file I/O
- Instead of hard-coding the initial values of matrices, read them from a file
- Instead of printing the product of the two matrices on the screen, print it to the file

References

- C Programming Language, Brian Kernighan and Dennis Ritchie
- Let Us C, Yashavant Kanetkar
- C for Dummies, Dan Gookin
- <http://cplusplus.com>
- <http://www.gidforums.com/t-4785.html>