



```
*****  
* convolve.c  
***** /
```

```
/* Standard includes */  
#include <assert.h>  
#include <math.h>  
#include <stdlib.h> /* malloc(), realloc() */
```

```
/* Our includes */  
#include "base.h"  
#include "error.h"  
#include "convolve.h"  
#include "klt_util.h" /* printing */
```

```
#define MAX_KERNEL_WIDTH 71
```

```
typedef struct {  
    int width;  
    float data[MAX_KERNEL_WIDTH];  
} ConvolutionKernel;
```

```
/* Kernels */
```

# Fundamentals of Programming

## session 19

### Passing arrays to functions, Sorting arrays

# Passing arrays to functions

---

```
#include <stdio.h>

void myPuts(char[]);

int main() {
    char s[100] = "Salam Kako!!";

    myPuts(s);

    return 0;
}

void myPuts(char s[]) {

    for (int i = 0; s[i] != '\0'; i++)
        putchar(s[i]);

    putchar('\n');
}
```

arrayfunc.c

# Passing arrays to functions

```
void printArray(int a[], int n);

int main() {
    int a[] = {1,2,3,4,5,6};

    printArray(a, 6);

    return 0;
}

void printArray(int a[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d, ", a[i]);

    putchar('\n');
}
```

arrayfunc2.c

# Passing arrays to functions

```
void printArray(int a[], int n);

int main() {
    int a[] = {1,2,3,4,5,6};

    printArray(a, 4);

    return 0;
}

void printArray(int a[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d, ", a[i]);

    putchar('\n');
}
```

# Passing arrays to functions

```
void printArray(int a[], int n);

int main() {
    int a[] = {1,2,3,4,5,6};
    printArray(a, 8);

    return 0;
}

void printArray(int a[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d, ", a[i]);

    putchar('\n');
}
```

# Passing arrays to functions

```
void printArray(int a[], int n);

int main() {
    int a[] = {1,2,3,4,5,6};

    printArray(a, 80000);

    return 0;
}

void printArray(int a[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d, ", a[i]);

    putchar('\n');
}
```

# Changing array elements inside function

```
#include <stdio.h>

void printArray(int[], int);
void changeArray(int[]);

int main() {
    int a[] = {1,2,3,4,5,6};

    printArray(a, 6);

    changeArray(a);

    printArray(a, 6);

    return 0;
}

void changeArray(int a[]) {
    a[0] = 8;
}

void printArray(int a[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d, ", a[i]);
    putchar('\n');
}
```

arrayfunc3.c

# const

```
#include <stdio.h>

int main() {
    const int a = 10;

    a += 10;
}
```

const.c



# const

```
#include <stdio.h>

int main() {
    const int a = 10;

    a += 10;
}
```

```
behrooz:code$ gcc const.c
const.c: In function 'main':
const.c:7:5: error: assignment of read-only variable 'a'
    a += 10;
    ^
```

const.c

# const for arrays (principle of least privilege)

```
#include <stdio.h>

void printArray(const int[], int);
void changeArray(int[]);

int main() {
    int a[] = {1,2,3,4,5,6};

    printArray(a, 6);

    changeArray(a);

    printArray(a, 6);

    return 0;
}

void changeArray(int a[]) {
    a[0] = 8;
}

void printArray(const int a[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d, ", a[i]);
    putchar('\n');
}
```

arrayfunc4.c

# const for arrays (principle of least privilege)

```
#include <stdio.h>

void printArray(const int[], int);
void changeArray(const int[]);

int main() {
    int a[] = {1,2,3,4,5,6};

    printArray(a, 6);

    changeArray(a);

    printArray(a, 6);

    return 0;
}

void changeArray(const int a[]) {
    a[0] = 8;
}

void printArray(const int a[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d, ", a[i]);
    putchar('\n');
}
```

arrayfunc5.c

# const for arrays (principle of least privilege)

```
#include <stdio.h>

void printArray(const int[], int);
void changeArray(const int[]);

int main() {
    int a[] = {1,2,3,4,5,6};

    printArray(a, 6);

    changeArray(a);

    printArray(a, 6);

    return 0;
}

void changeArray(const int a[]) {
    a[0] = 8;
}

void printArray(const int a[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d, ", a[i]);
    putchar('\n');
}
```

arrayfunc5.c

# const for arrays (principle of least privilege)

```
#include <stdio.h>

void printArray(const int[], int);
void changeArray(const int[]);

int main() {
    int a[] = {1,2,3,4,5,6};

    printArray(a, 6);

    changeArray(a);

    printArray(a, 6);

    return 0;
}

void changeArray(const int a[]) {
    a[0] = 8;
}

void printArray(const int a[], int n) {
    for (int i = 0; i < n; i++)
        printf("%d, ", a[i]);
    putchar('\n');
}
```

```
behrooz:code$ gcc arrayfunc5.c
arrayfunc5.c: In function 'changeArray':
arrayfunc5.c:19:8: error: assignment of read-only location '*a'
    a[0] = 8;
    ^
```

arrayfunc5.c

# Sorting Arrays

```
#define SIZE 11
```

```
int array[11] = {25, 19, 14, 17, 27, 6, 18, 20, 1, 21, 32};
```

25	19	14	18	27	6	32	18	20	1	21
----	----	----	----	----	---	----	----	----	---	----

# Sorting Arrays

```
#define SIZE 11
```

```
int array[11] = {25, 19, 14, 17, 27, 6, 18, 20, 1, 21, 32};
```

25	19	14	18	27	6	32	18	20	1	21
----	----	----	----	----	---	----	----	----	---	----

1	6	14	18	18	19	20	21	25	27	32
---	---	----	----	----	----	----	----	----	----	----

# Sorting Arrays

```
void printArray(int a[], int n);
void sort(int a[], int n);

int main() {
    int a[] = {25, 19, 14, 18, 27, 6, 32, 18, 20, 1, 21};
    int n = sizeof(a) / sizeof(a[0]);

    printArray(a,n);

    sort(a,n);

    printArray(a,n);

    return 0;
}

void sort(int a[], int n) {
    // ??
}
```

sortarray.c



# Sorting Arrays

```
void sort(int a[], int n) {  
    int m;  
  
    m = n-1;  
    for (int i = 0; i < m; i++) {  
  
        if (a[i] > a[i+1]) {  
            int temp = a[i];  
            a[i] = a[i+1];  
            a[i+1] = temp;  
        }  
  
    }  
  
}
```

sortarray2.c

25

19

14

18

27

6

32

18

20

1

21

# Sorting Arrays

```
void sort(int a[], int n) {  
    int m;  
  
    m = n-1;  
  
    for (int j = 0; j < m; j++) {  
        for (int i = 0; i < m; i++) {  
            if (a[i] > a[i+1]) {  
                int temp = a[i];  
                a[i] = a[i+1];  
                a[i+1] = temp;  
            }  
        }  
    }  
}
```

sortarray2.5.c

25

19

14

18

27

6

32

18

20

1

21

# Sorting Arrays

```
void sort(int a[], int n) {
    int m;

    for (m = n-1; m > 0; m--) {
        for (int i = 0; i < m; i++) {

            if (a[i] > a[i+1]) {
                int temp = a[i];
                a[i] = a[i+1];
                a[i+1] = temp;
            }

        }
    }
}
```

sortarray3.c

25

19

14

18

27

6

32

18

20

1

21

# Sorting Arrays

```
void sort(int a[], int n) {  
    int m;  
  
    for (m = n-1; m > 0; m--) {  
        for (int i = 0; i < m; i++) {  
  
            if (a[i] > a[i+1]) {  
                int temp = a[i];  
                a[i] = a[i+1];  
                a[i+1] = temp;  
            }  
  
        }  
    }  
}
```

sortarray3.c

descending sort?

25

19

14

18

27

6

32

18

20

1

21

# Mean, median and mode

Write a program printing mean, median, and mode of an arrays of integers. You can assume that array elements are integers between 0 and 19.

```
void printArray(const int a[], int n);
void sort(int a[], int n);
double mean(const int a[], int n);
double median(int a[], int n);
int mode(const int a[], int n);

int main() {
    int a[] = {1,18,2,3,4,5,7,7,4,19,18,12,13, 0, 18,
               14,17,1,0,8,9,9,9,10, 11, 0,19,18,
               18,1,3,4,6,7,8,3,2,15,1,0,7,13,14,
               10,12,16,17,18,1,4,6,8,4,0,5,8,7,5,
               6,4,9,16,15,12,13,14,12,13,16,18,9,10};

    int n = sizeof(a) / sizeof(a[0]);

    printArray(a,n);

    printf("mean=%.2f, mode= %d, median=%.1f\n", mean(a,n), mode(a,n), median(a,n));

    return 0;
}
```

stats.c

# Mean, median and mode

Write a program printing mean, median, and mode of an arrays of integers. You can assume that array elements are integers between 0 and 19.

```
double mean(const int a[], int n) {  
  
    int sum = 0;  
    for (int i = 0; i < n; i++)  
        sum += a[i];  
  
    return sum / (double) n;  
}
```

stats.c