Array example and initialization

```c
#include <stdio.h>
#define size 10
int main () {
   int n1[10];
   int i;
   for( i = 0; i < 10; i = i + 1 ){  n1[i] = 0;  }

   printf("%s%13s\n", "Elements", "Value");
   for( i = 0; i < 10; i = i + 1 ) printf("%7d%13d\n", i, n1[i]);

   int n2[size] = {30, 28, 47, 52, 42, 44, 48, 78, 92, 10 };
   printf("%s%13s\n", "Elements", "Value");
   for( i = 0; i < size; i = i + 1 ) printf("%7d%13d\n", i, n2[i]);

   int n3[size] = {11, 22, 33, 44, 55};
   printf("%s%13s\n", "Elements", "Value");
   for( i = 0; i < size; i = i + 1 ) printf("%7d%13d\n", i, n3[i]);
   return 0;
}
```

Histogram/plot printing program with array

```c
#include <stdio.h>
#define size 10

int main () {
   int n[size] = {15, 10, 5, 7, 11, 13, 1, 5, 15, 10 };
   int i, j;

   printf("Element        Value        Histogram\n");

   for( i = 0; i < size; i = i + 1 ){
      printf("%7d%13d", i, n[i]);
      for( j = 0; j <= n[i]; j = j + 1 ) printf("%c", '|');
      printf("\n");
   }

   return 0;
}
```

Treating character array as strings

```c
#include <stdio.h>
int main () {
   char s1[20] ;
   char s2[] = "string literal";
   /* here s2 reserve 15 characters; 14 are given above
      and it ends with a character '\0' known as null character */
   printf("Enter a string: ");
   scanf("%s", s1);
   printf("First string: %s\nSecond string: %s\n"
          "Given string with spaces between characters is:\n", s1, s2);
   int i ;
   for( i = 0; s1[i] != '\0'; i = i + 1 ){
      printf("%c ", s1[i]);
   }
   printf("\n");
   return 0;
}
```

Summarizing the results of rolling die in an array

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define size 6
int main () {
   int face, roll, frequency[size] = {0};
   srand( time(NULL));
   for( roll = 1; roll <= 6000; roll = roll + 1 ){
      face = 1 + rand()%6;
      ++frequency[face];
   }
   printf("Face        Frequency\n");
   for( face = 1; face < size; face = face + 1 ){
      printf("%4d%17d\n", face, frequency[face]);
   }
   return 0;
}
```

Static local arrays and automatic local arrays

```
#include <stdio.h>
void staticArrayInit();

void automaticArrayInit();

int main () {
   printf("First call to each function");
   staticArrayInit();
   automaticArrayInit();

   printf("\n\nSecond call to each function");
   staticArrayInit();
   automaticArrayInit();

   printf("\n");
   return 0;
}

void staticArrayInit(){
   static int array1[3];
```

```
void staticArrayInit(){
   static int array1[3];
   int i;
   printf("\nValues of entering staticArrayInit:\n");
   for( i = 0; i <= 2; i++ ){
      printf("array1[%d] = %d ", i, array1[i]); }
   printf("\nValues on exiting staticArrayInit:\n");
   for( i = 0; i <= 2; i++ ){
      printf("array1[ %d ] = %d ", i, array1[i] += 5);
   }
}
void automaticArrayInit(){
   static int array2[3] = { 1, 2, 3 };
   int i;
   printf("\n\nValues on entering automaticArrayInit:\n");
   for( i = 0; i < 3; i++ ){
      printf("array2[%d] = %d ", i, array2[i]);
   }
   printf("\nValues on exiting automaticArrayInit:\n");
   for( i = 0; i < 3; i++ ){
      printf("array2[ %d ] = %d ", i, array2[i] += 5);
   }
}
```

Passing arrays and individual array elements to functions

```
#include <stdio.h>
#define size 5
void modifyArray(int [], int);
void modifyElement(int);
int main () {
   int a1[size] = {0, 1, 2, 3, 4 };
   int i;
   printf("Effects of passing entire array by reference:\n\nThe "
          "values of the original array are:\n");
   for( i = 0; i < size; i = i + 1 ) printf("%3d", a1[i]);
   printf("\n");

   modifyArray(a1, size);

   printf("The values of the modified array are:\n");
   for( i = 0; i < size; i = i + 1 ) printf("%3d", a1[i]);
   printf("\n\n\nEffects of passing array element "
          "by value:\n\nThe value of a[3] is %d\n", a1[3]);
```

```
   printf("\n\n\nEffects of passing array element "
          "by value:\n\nThe value of a[3] is %d\n", a1[3]);
   modifyElement(a1[3]);
   printf("The value of a[3] is %d\n", a1[3]);
   return 0;
}
void modifyArray(int b[], int x){ /*here "int size" can conflict variables.*/
   int j;
   for( j = 0; j < x; j++ ) b[j] *= 2;
}
void modifyElement(int e){
   printf("Value in modifyElement is %d\n", e *= 2 );
}
```

Demonstrating the const type qualifier with arrays

```c
#include <stdio.h>
void tryToModifyArray(const int []);
int main () {
   int a2[] = {10, 20, 30};
   tryToModifyArray(a2);
   printf("%d %d %d\n", a2[0], a2[1], a2[2]);
   return 0;
}

void tryToModifyArray(const int b[]){
   b[0]/ = 2;
   b[1]/= 2;
   b[2]/= 2;
}
```

This program will not compile. There are two errors. Given operation `/ =` is not defined because there is a space between slash and equal. Another problem in compilation is: we are trying to modify constant array.

Sorting an array's values into ascending order

```
#include <stdio.h>
#define size 10
int main () {
   int a1[size] = {2, 5, 6, 18, 3, 28, 99, 20, 21, 35 };
   int p, i, temp;
   printf("Data items in original order\n");
   for( i = 0; i < size; i++ ) printf("%4d", a1[i]);

   /* bubble sorting */
   for( p = 0; p < size; p++ ){
      for( i = 0; i < size - 1; i++ ){
         if( a1[i] > a1[i+1] ){
            temp = a1[i];
            a1[i] = a1[i+1];
            a1[i+1] = temp;
         }}
   }
   printf("\nData items in ascending order\n");
   for( i = 0; i < size; i++ ) printf("%4d", a1[i]);        printf("\n");
   return 0;
}
```

Data analysis

```
/*
  File name: lecture10_5.c
  This program introduces the topic of survey data analysis.
  It computes the mean, median and mode of the data
  Linear search of an array
  Binary search of a sorted array
*/
#include <stdio.h>
#define size 100
#define size1 15
void mean(const int []);
void median(int []);
void mode(int [], const int []);
void bubbleSort(int []);
void printArray(const int []);
int linearSearch(const int [], int, int);
int binarySearch(const int [], int, int, int);
void printHeader();
void printRow(const int [], int, int, int);
int main(){
  int f[10]={0};
  int r[size] =
```

```
int main(){
  int f[10]={0};
  int r[size] =
    {6, 7, 8, 9, 8, 7, 8, 9, 8, 9,
     7, 8, 9, 5, 9, 8, 7, 8, 7, 8,
     6, 7, 8, 9, 3, 9, 8, 7, 8, 7,
     7, 8, 9, 8, 9, 8, 9, 7, 8, 9,
     6, 7, 8, 7, 8, 7, 9, 8, 9, 2,
     7, 8, 9, 8, 9, 8, 9, 7, 5, 3,
     5, 6, 7, 2, 5, 3, 9, 4, 6, 4,
     7, 8, 9, 6, 8, 7, 8, 9, 7, 8,
     7, 4, 4, 2, 5, 3, 8, 7, 5, 6,
     4, 5, 6, 1, 6, 5, 7, 8, 7, 5};
  mean(r);
  median(r);
  mode(f, r);

  /* creating new data */
  int a1[size], x, searchKey, element;
  for(x=0; x< size; x++)a1[x]=2*x;
  printf("Enter integer search key:\n");
  scanf("%d", &searchKey);
  element = linearSearch(a1, searchKey, size);
  if(element != -1) printf("Found value in element %d\n", element);
```

```
  if(element != -1) printf("Found value in element %d\n", element);
  else printf("Value not found!\n");

  /* re-creating new data */
  int a2[size1], key, result;
  for(x=0; x< size; x++)a2[x]=2*x;
  printf( "Enter a number between 0 and 28: " );
  scanf( "%d", &key );
  printHeader();
  result = binarySearch(a2, key, 0, size1 - 1 );
  if(result != -1) printf("\n%d found in array element %d\n", key, result );
  else printf("\n%d not found\n", key);
  return 0;
}
/* calculate average of all response values */
void mean(const int answer[]){
  int j, total=0;
  printf("\n********\n Mean \n********\n");
  for(j=0; j< size; j++) total+=answer[j];
  printf("Mean is the average value of the data item.\n"
     "Mean value of (%d) data items is: %d/%d= %.3f\n\n",
     size, total, size, (float) total/size);
}
void median(int answer[]){
```

```
void median(int answer[]){
  printf("\n********\n Median \n********\n"
     "Unsorted array: \n");
  printArray(answer);
  bubbleSort(answer);
  printf("\n\nThe sorted array is");
  printArray(answer);
  printf("\n\nThe median is %dth element of sorted %d element array.\n"
     "Median is %d\n\n", size/2, size, answer[size/2]);
}
void mode(int freq[], const int answer[]){
  int rate, j, histo, largest=0, modeValue=0;
  printf("\n********\n  Mode\n********\n");
  for(rate=1; rate<=9; rate++)freq[rate]=0;
  for(j=0; j< size; j++) ++freq[answer[j]];
  printf("Response  Frequency          Histogram:\n\n");
  for(rate=0; rate<=9; rate++){
    printf( "%8d%11d ", rate, freq[rate]);
    if(freq[rate]>largest){
      largest=freq[rate];
      modeValue =rate;
    }
    for(histo=1; histo<=freq[rate]; histo++)printf("*");
    printf("\n");
```

```
    for(histo=1; histo<=freq[rate]; histo++)printf("*");
    printf("\n");
  }
  printf("Mode is most frequent value.\n"
      "Here mode %d occurred %d times.\n",
      modeValue, largest);
}
void bubbleSort(int a[]){
  int p, j, temp;
  for(p=1; p< size; p++)
    for(j=0; j< size-1; j++)
      if(a[j]>a[j+1]){
    temp=a[j];
    a[j]=a[j+1];
    a[j+1]=temp;
      }
}
void printArray(const int a[]){
  int j;
  for(j=0; j< size; j++){
    if(j%20 == 0) printf("\n");
    printf("%2d", a[j]);
  }
}
```

```
}


int linearSearch( const int array[], int key, int x){
  int n;
  for(n=0; n< x; ++n)
    if(array[n]==key)return n;
  return -1;
}
int binarySearch( const int b[], int searchKey, int low, int high ){
  int middle;
  while(low<=high){
    middle = (low + high)/2;
    printRow( b, low, middle, high );
    if(searchKey == b[middle]) return middle;
    else if(searchKey < b[middle])high = middle-1;
    else low = middle+1;
  }
  return -1;
}
void printHeader(){
  int i;
  printf("\nSubscripts:\n");
  for(i=0; i< size1; i++)printf("%3d ",i);
```

```
  printf("\n");
  for(i=0; i<4*size1; i++)printf("-");
  printf("\n");
}
void printRow(const int b[], int low, int mid, int high ){
  int i;
  for(i=0; i< size1; i++)
    if(i< low || i>high)printf("    ");
    else if(i==mid)printf("%3d*", b[i]);
    else printf("%3d ",b[i]);
  printf("\n");
}
```