##### Chapter 7 Arrays

1. If b is an array with type int elements, then the statement

b += 5;

adds five to each element of b. [False]

1. If b is an array with type int elements and the value of b[4] is 3, then the statement

printf("%d\n", b[b[4] - 1]);

displays one less than the value of b[3]. [False]

1. If b is an array of integer elements, then the statement

b[3] \*= 2;

doubles the value of b[3]. [True]

1. What value is returned by function result?

int

result(const int a[], int n)

{

int i, r;

r = 0;

for (i = 1; i < n; ++i)

if (a[i] > a[r])

r = i;

return (r);

}

\*a. The subscript of the largest of the first n elements of array a.

b. The value of the largest of the first n elements of array a.

c. The subscript of the smallest of the first n elements of array a.

d. The value of the smallest of the first n elements of array a.

e. The subscript of the last element greater than its predecessor within the first n elements of array a.

1. What is the effect of this program segment?

#define MAX 50

int a[MAX], i, j, temp;

for (i = 0; i < MAX / 2; ++i) {

temp = a[i];

a[i] = a[MAX - i - 1];

a[MAX - i - 1] = temp;

}

a. Arranges the elements of array a in ascending order.

b. Counts the number of elements of a greater than its first element.

\*c. Reverses the numbers stored in the array.

d. Puts the largest value in the last array position.

e. None of the above.

1. What is the effect of the following program segment?

#define MAX 50

int a[MAX], i, j, temp;

for (i = 0; i < MAX - 1; ++i)

if (a[i] > a[i + 1]) {

temp = a[i];

a[i] = a[i + 1];

a[i + 1] = temp;

}

a. Arranges the elements of array a in ascending order.

b. Counts the number of elements of a greater than its first element.

c. Reverses the numbers stored in the array.

\*d. Puts the largest value in the last array position.

e. None of the above.

1. What is the effect of the following program segment?

#define MAX 50

int a[MAX], i, j, temp;

temp = 0;

for (i = 1; i < MAX; ++i)

if (a[i] > a[0])

++temp;

a. Arranges the elements of array a in ascending order.

\*b. Counts the number of elements of array a greater than its initial element.

c. Reverses the numbers stored in the array.

d. Puts the largest value in the last array position.

e. None of the above.

1. What is the maximum valid subscript value for array a?

#define MAX 50

int a[MAX], i, j, temp;

a. 0

\*b. 49

c. 50

d. a[50]

e. none of the above

1. What is the minimum valid subscript value for array a?

\*a. 0

b. 1

c. any negative number

d. There is no minimum.

e. none of the above

1. When a C program passes an array as a function argument,

a. the value of the initial element of the array is actually passed.

\*b. the address of the initial element of the array is actually passed.

c. the entire array is copied into the function's data area.

d. the addresses of the initial and final elements of the array are copied into the function's data area.

e. none of the above.

1. How would you best describe the purpose of the following code?

f = 0;

for (i = 1; i < N; ++i)

if (a[i] >= a[f])

f = i;

a. Rearrange the first N components of array a in descending order.

b. Rearrange the first N components of array a in ascending order.

c. Place the largest component of array a in position N.

d. Compute the value of the largest component in array a.

\*e. Determine the subscript of the last occurrence of the largest of the first N components of array a.

1. How many numbers can be stored in the array declared below?

double arr[10][5][6];

a. 21

b. 90

c. 180

\*d. 300

e. none of the above