C Arrays (1D) - Answers

**Pre lab Questions**

**Question1:** Write statements to accomplish each of the following:

a) Display the value of the seventh element of character array f.

**ANS:** printf( “%c\n”, f[ 6 ] );

b) Input a value into element 4 of single-subscripted floating-point array b.

**ANS:** scanf( “%f”, &b[ 4 ] );

c) Initialize each of the 5 elements of single-subscripted integer array g to 8.

**ANS:**

for ( loop = 0; loop <= 4; loop++ )

g[ loop ] = 8;

d) Total the elements of floating-point array c of 100 elements.

**ANS:**

for ( loop = 0; loop <= 99; loop++ )

sum += c[ loop ];

e) Copy array a into the first portion of array b. Assume double a[ 11 ], b[ 34 ];

**ANS:**

for ( loop = 0; loop <= 10; loop++ )

b[ loop ] = a[ loop ];

f) Determine and print the smallest and largest values contained in 99-element floating-point array w.

**ANS:**

smallest = largest = w[ 0 ];

for ( loop = 1; loop <= 98; loop++ )

if ( w[ loop ] < smallest )

smallest = w[ loop ];

else if ( w[ loop ] > largest )

largest = w[ loop ];

**Question2:** Write single statements that perform each of the following single-subscripted array operations:

a) Initialize the 10 elements of integer array counts to zeros.

**ANS:**

for ( i = 0; i <= 9; i++ )

counts[ i ] = 0;

b) Add 1 to each of the 15 elements of integer array bonus.

**ANS:**

for ( i = 0; i <= 14; i++ )

++bonus[ i ];

c) Read the 12 values of floating-point array monthlyTemperatures from the keyboard.

**ANS:**

for ( i = 0; i <= 11; i++ ) {

printf( “Enter a temperature: ” );

scanf( “%f”, &monthlyTemperatures[ i ] );

}

d) Print the 5 values of integer array bestScores in column format.

**ANS:**

for ( i = 0; i <= 4; i++ ) {

printf( “%d\t”, bestScores[ i ] );

**Question3:** Find the error(s) in each of the following statements:

a) Assume**:** int a[ 3 ];

printf( "$d %d %d\n", a[ 1 ], a[ 2 ], a[ 3 ] );

**ANS:** printf( “%d %d %d\n”, a[ 0 ], a[ 1 ], a[ 2 ] );

b) double f[ 3 ] = { 1.1, 10.01, 100.001, 1000.0001 };

**ANS:** Too many variables defined.

double f[ 3 ] = { 1.1, 10.01, 100.01 };

**Lab Tasks**

**1-** Write a C program that does the multiplication of numbers start from 5 to 35 incrementing by 5 and store them in a 1D array having 7 elements.

product= 5 \* 10 \* 15 \* 20 \* 25 \* 30 \*35

#include"stdio.h"

#define SIZE 7

main()

{

 int x[SIZE],num=5;

 long product=1;

/\*read into the array\*/

 for(int i=0;i<SIZE;i++)

 {

 x[i]=num;

 num+=5;

 }

/\*calculate the product\*/

 for(int j=0;j<S;j++)

 product\*=x[j];

 printf("product = %ld",product);

}

2- Write a C program that prompts the user the enter 5 numbers and display the numbers that divisible with two.

#  **SAMPLE RUN**

 Enter number 1:**4**

 Enter number 2:**3**

 Enter number 3:**5**

 Enter number 4:**8**

 Enter number 5:**6**

 number[0]= 4

 number[3]= 8

 number[4]= 6

**answer:**

#include"stdio.h"

#define SIZE 5

main()

{

 int x[SIZE],num=5;

 for(int i=0;i<SIZE;i++)

 {

 printf("Enter number %d : ",i+1);

 scanf("%d",&x[i]);

 }

 for(int j=0;j<S;j++)

 if(x[j]%2==0)

 printf("\n number[%d]= %d",j,x[j]);

}

3- Write a C program that accepts student numbers and midterm results for 5 students from the user, and calculates and displays the average of the midterms as well as the student numbers and their corresponding midterms that are above the average as shown in the SAMPLE RUN. The student numbers should be long integers, the midterm results should be integers, and the average should be a real number having 2 numbers after the decimal point. Holds the student numbers and midterm results in a different arrays as a character. (use 1D array)

SAMPLE RUN:

Enter student number and midterm result for 5 students:

991234 45

993456 78

018899 100

019988 56

029999 85

The Average is 72.80

The students who are above the average are:

993456 78

018899 100

029999 85

**answer:**

#include"stdio.h"

#define SIZE 5

main()

{

 long students[SIZE];

 int i, sum=0, midterms[SIZE];

float ave;

 printf("Enter student number and midterm result for %d students:\n", SIZE);

 for (i=0; i<SIZE; ++i)

 {

 scanf("%ld%d", &students[i], &midterms[i]);

 sum += midterms[i];

 }

ave = (float)sum/SIZE;

 printf("\nThe Average is %.2f",ave);

 printf("\nThe students who are above the average are:") ;

 for (int i=0; i<SIZE; ++i)

 {

 if(midterms[i] > ave)

 printf("\n%6ld %3d",students[i],midterms[i]);

 }

}

**4-** Write a program that reads five numbers into an 1D array(each between 1 and 30). For each number read, your program should print a line containing that number of adjacent asteriks. For example, if your program reads the number seven, it should print \*\*\*\*\*\*\*

(you can assign a random number for the array)

**answer:**

#include<stdio.h>

#define SIZE 5

main()

{

 int x,arrayX[SIZE],num;

 for(x=0;x<SIZE;x++)

 {

 printf("enter %d.number: ",x+1);

 scanf("%d",&num);

 if( (num>=1) && (num<=30) )

 arrayX[x]=num;

 else

 {

 printf("\nYou entered a wrong number.\nPlease try again\n");

 x--;

 }

 }

 for(x=0;x<S;x++)

 {

 for(y=1;y<=arrayX[x];y++)

 printf("\*");

 printf("\n");

 }

}

**5-** Write a program that displays a table of integers with their squares and cubes as shown below. Integer values should change from -5 to 4 with an increment of 1 and should be initialized in a 1D array having 10 elements. Square and cubes values should also be stored in a 1D array having 10 elements. Use the following formula and for loop to generate the square and cube values.

 X2 for x<0

y =

 X3 for x>=0

Output

x y

--------

-5 25

-4 16

-3 9

-2 4

-1 1

 0 0

 1 1

 2 8

 3 27

 4 64

**answer:**

#include"stdio.h"

#include"math.h"

#define SIZE 10

main()

{

 int x[SIZE],num=-5;

 for(int i=0;i<SIZE;i++)

 {

 x[i]=num;

 num++;

 }

 int arrayY[10],j;

 for(j=0;j<SIZE;j++)

 {

 if(x[j]<0)

 arrayY[j]=pow(x[j],2);

 else

 arrayY[j]=pow(x[j],3);

 }

 printf("%2s%3s\n","x","y");

 printf("-----\n");

 for(j=0;j<SIZE;j++)

 printf("%2d%3d\n",x[j],arrayY[j]);

}

**6-** Write a program that displays a table of integers with their squares and cubes as shown below. X values should be initialized in a 1D array having 10 elements. Y values should also be stored in a 1D array having 10 elements. Use the following formula and for loop to generate the square and cube values. (assign random number for array X)

 X2 for even X value

y =

 X3 for odd X value

Output

x y

--------

10 100

 3 27

 5 125

12 144

 8 64

 6 36

 7 343

 4 16

14 196

 9 729

**answer:**

#include"stdio.h"

#include"math.h"

#include"stdlib.h"

#define SIZE 10

main()

{

 int arrayX[SIZE],num;

 randomize();

 for(int i=0;i<SIZE;i++)

 {

 num=3+rand()%12; /\*random number from 3-14\*/

 arrayX[i]=num;

 }

 int arrayY[10],j;

 for(j=0;j<SIZE;j++)

 {

 if(arrayX[j]%2==0)

 arrayY[j]=pow(arrayX[j],2);

 else

 arrayY[j]=pow(arrayX[j],3);

 }

 printf("%3s%7s\n","x","y");

 printf("----------\n");

 for(j=0;j<SIZE;j++)

 printf("%3d%7d\n",arrayX[j],arrayY[j]);

}