Practice 2.2: Explain the difference amongst the following 3 functions:

void swap1 (int x, int y)

{

int temp = x;

x = y;

y = temp;

}

void swap2(int &x, int &y)

{

int temp = x

x = y;

y = temp;

}

void swap3 (int \*x, int \*y)

{

int temp = \*x;

\*x = \*y;

\*y = temp;

}

Ans:

* In the first function, the values ​​of x and y are copied into the function and interacted on it. The two variables will not be involved any changes occuring in the function. When the function ends, the value of each variable remains the same.
* In the second function, the address of the two variables are passed to the function, which means they are involved the changes that occur in the function. When the function ends, one variable will get the value of other variable and vice versa.
* In the third function, \*x and \*y are pointer x and pointer y. Two pointers are passed to function, which means which means they are involved the changes that occur in the function. When the function ends, one variable will get the value of other variable and vice versa.

Practice 2.3: Given the following program:

void main()

{

double m[100];

double \*p1, \*p2;

p1 = m;

p2 = &m[6];

}

How many bytes between p1 and p2?

Ans: p1 will get the address of m[0] and p2 will get the address of m[6]. There are 4 bytes between m[0] and m[1]. So between p1 and p2, there are: 4\*6=24 bytes.

Practice 2.4: What is displayed on screen of the following code:

#include <stdio.h>

void main()

{

int x = 1023;

char \*p = (char \*)&x;

printf (“%d %d %d %d\n”, p[0], p[1], p[2], p[3]);

}

Ans: The value of p[0], p[1], p[2], p[3] is displayed on screen in this order:

-1 3 0 0.