Name: Reno Redaja

Date: October 19, 2020

1. Declare a string using the pointer notation. Initialize to the value “EMPTY”.

char \*c = “EMPTY”;

1. Declare an array of characters. This array will be 10 cells large. Initialize to any combination of upper-case alpha-characters.

char str[] = {‘A’, ‘E’, ‘I’, ‘O’, ‘U’, ‘R’, ‘G’, ‘B’, ‘Y’, ‘Q’};

1. Using the array in question 2 above, write a module that converts all upper-case values to lower case. Assume that the array holds only alpha-characters, and that they are all upper case. (Subtract 32 from the upper case)

void toLowerCase(char c[]) {

int len;

len = strlen(c);

for (int i = 0; i < len; i++) {

if (c[i] >= ‘A’ && c[i] <= ‘Z’) {

c[i] = c[i] + 32;

} //end inner if statement

} //end for loop

} //end toLowerCase

1. Given the following declaration, show the effect. Each command runs separately from the rest.

Int X[6]; // each integer is 16 bits long, therefore 2 memory cells per integer.

X

25 14 54 65 12 8

[0] [1] [2] [3] [4] [5]

ADDRESS 8500 8502 8504 8506 8508 8510

1. \*Xptr = 6;

assign 6 to index at X[0]

1. \*X = 3;

dereference X[0] to 3

1. X[3] = 24;

assign 24 to index at X[3]

1. Xptr[4] = 3

assign 3 to index at X[4]

1. \*(X +4) = 2;

assign 2 to index at X[4]

1. \*(Xptr + 3) = 32;

assign 32 to index at X[3]

1. Xptr +=3;

increment Xptr 3 times, Xptr is pointing to index at X[3]

1. Xptr++; Y = Xptr - X; // two commands for this one.

increment Xptr 1 time, Xptr is pointing to index at X[1] then Y is assigned to 1. Since there are 2 memory cells per integer, we have to subtract X[1] from X[0] therefore 1-0 = 1.