|  |  |  |
| --- | --- | --- |
| American University of SharjahCollege of Engineering Dept of Computer Science & Engg  P. O. Box 26666  Sharjah, UAE |  | Instructors: Dr. Rana Ahmed **Lab Instructor:** Eng. Sameer Alawnah  **Office: EB2-101**  **Phone**: +97165152974  **e-mail**: salawnah@aus.edu  **Semester**: Summer 2016 |

**CMP 120L– Introduction to Computer Science I Lab**

**Lab 12**

**Exercise 1:**

Write a program that declares and initializes three integer variables **x, y,** and **z** with some integer values. It then declares a pointer **ptr** to an integer variable. Using **ptr** , address of (**&**) and dereferencing (**\***) operators, increment each of the variables **x, y,z**, by one and print those updated variables.

#include <iostream>

using namespace std;

void main()

{

int x, y, z;

int \*ptr;

x = 2;

y = 4;

z = 7;

ptr = &x;

(\*ptr)++;

cout << "x = " << \*ptr << endl;

ptr = &y;

(\*ptr)++;

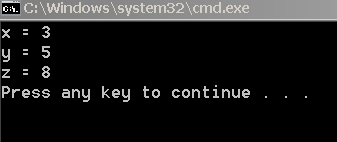
cout << "y = " << \*ptr << endl;

ptr = &z;

(\*ptr)++;

cout << "z = " << \*ptr << endl;

}



**Exercise 2:**

Write a main program that declares and initializes an integer array **z** of 5 integer elements. Initialize the elements of **z** array with some integer values. The program performs the following tasks (in this sequence):

1. Declare a pointer **p** to an integer, and point it to the last element of the array **z**.
2. Print the address of the last element of **z** array.
3. By incrementing (or decrementing) the pointer **p** appropriately, and using dereferencing operator, print the contents of **z** array in the reverse order (i.e., first z[4], then z[3],… z[0]).
4. Print the starting address of the array **z**.
5. Print the address of element z[1].

#include <iostream>

using namespace std;

void main()

{

int z[5] = { 1, 2, 3, 4, 5 };

int \*p;

p = &z[4];

cout << "Address of the last element of z array is:" << p << endl;

for (int i = 4; i >= 0; i--)

{

cout << "z[" << i << "] = " << \*p << endl;

p--;

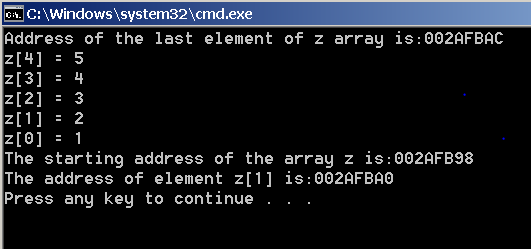
}

cout << "The starting address of the array z is:" << p << endl;

p = &z[1];

cout << "The address of element z[1] is:" << p << endl;

}



**Exercise 3:**

Write a program that declares and initializes a character array **z** of size 5 with characters ‘H’, ‘E’, ‘L’, ‘L’, and ‘O’. It then declares a pointer **ptr** to a character variable. Using **ptr** , address of (**&**) and dereferencing (**\***) operators, your program should count how many vowels are present in the array **z,** and print this count. Assume only uppercase characters are present in the array. **Note**: Vowels are characters ‘A’, ‘E’, ‘I’, ‘O’, and ‘U’.

#include <iostream>

using namespace std;

void main()

{

char z[5] = { 'H', 'E', 'L', 'L', 'O' };

char \*ptr;

ptr = &z[0];

int count = 0;

for (int i = 0; i < 5; i++)

{

if ((\*ptr == 'A') || (\*ptr == 'E') || (\*ptr == 'I') || (\*ptr == 'O') || (\*ptr == 'U'))

{

count++;

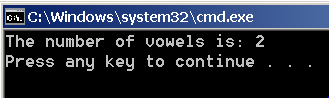
}

ptr++;

}

cout << "The number of vowels is: " << count << endl;

}



**Exercise 4:**

Write a main program that declares and initializes a double array **d** of 5 double elements. Initialize the elements of **d** array with some double values. The program then prints the address of each element of **d** array (i.e., addresses of d[0], d[1]…, d[4]).

By observing two consecutive addresses on the output, find the number of bytes needed to store a double number. This last step should be done without writing any code.

**Hint:**

For the last step, you may need to use calculator supporting hexadecimal arithmetic. You may use the

following online hexadecimal calculator to find difference between two hexadecimal numbers.

<http://www.miniwebtool.com/hex-calculator/>

#include <iostream>

using namespace std;

void main()

{

double d[5] = { 1.2, 1.3, 1.4, 1.5, 1.6 };

double \*ptr;

ptr = &d[0];

for (int i = 0; i < 5; i++)

{

cout << "d[" << i << "] = " << ptr << endl;

ptr++;

}

}

