|  |  |  |
| --- | --- | --- |
| 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 11**

**Exercise 1:**

Write a C++ function **void absolute (int x[ ], int N);** that accepts an integer array **x** of size **N**, and updates each of its element with its absolute value. For example, if the passed array to f1() is int p[5]= {11,-3,20,-9}; then f1(p,5) should update the p array to {11,3,20,9} in the main()..

Write a main() program to test the working of your function All printouts should be done in main()..

#include <iostream>

#include <cmath>

using namespace std;

void absolute(int x[], int N);

void main()

{

int i, x[5] = { 11, -3, 20, -9 };

cout << "array before absolue value:" << endl;

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

{

cout << x[i] << endl;

}

absolute(x, 5);

cout << "updated array with absolute value: " << endl;

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

{

cout << x[i] << endl;

}

}

void absolute(int x[], int N)

{

int i;

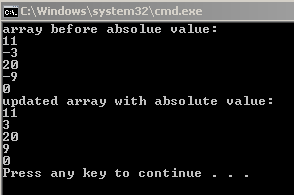
for (i = 0; i < 5; i++)

{

x[i]= abs(x[i]);

}

}



**Exercise 2:**

Write a function that takes the following prototype:

**void add\_vectors(int a[], int b[], int c[], int N);**

The function adds the ith elements of two one-dimensional arrays **a** and **b** ( each of size **N**)to produce the ith element of **c** array**;** i.e.,

c[i] = a[i]+b[i] for i=0,1,.., N-1.

Test the function with a main program that defines two arrays of 5 integers and initializes them within the main program.

***Sample Session:***

/\*

In main(); declare and initialize the following two arrays:

int x[5]= {10,20,30,40,50};

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

int z[5];

After a call to the function add\_vector(x, y, z, 5) in the main, the array z printed in the main() should be {15, 24,33,42,51}.

\*/

#include <iostream>

using namespace std;

void add\_vectors(int a[], int b[], int c[], int N);

void main()

{

int i;

int x[5] = { 10, 20, 30, 40, 50 };

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

int z[5];

cout << "vector a = " << endl;

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

{

cout << x[i] << endl;

}

cout << "vector b = " << endl;

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

{

cout << y[i] << endl;

}

add\_vectors(x, y, z, 5);

cout << "vector z = " << endl;

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

{

cout << z[i] << endl;

}

}

void add\_vectors(int a[], int b[], int c[], int N)

{

int i;

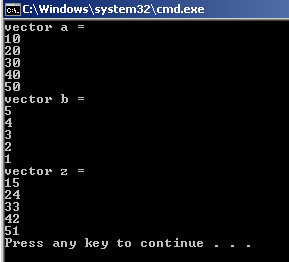
for (i = 0; i < 5; i++)

{

c[i]= a[i]+b[i];

}

}



**Exercise 3:**

Write and test the following function:

**void reverse(int a[ ], int N);**

The function reverses the **a[ ]** array of size **N**. For example, if in main(), we have int x[5]={1,20,3,4,5}; then a call reverse(x,5) will change x array to {5,4,3,20,1};

**Hint:** You may need to call swap() function within the code of reverse().

Write a suitable main program to test the working of reverse() function.

#include <iostream>

using namespace std;

void reverse(int a[], int N);

void main()

{

int i;

int x[5] = { 1, 20, 3, 4, 5 };

cout << "original array:" << endl;

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

{

cout << x[i] << endl;

}

reverse(x, 5);

cout << "reversed array: " << endl;

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

{

cout << x[i] << endl;

}

}

void reverse(int a[], int N)

{

int i, j;

for (i = 0; i < (N/2); i++)

{

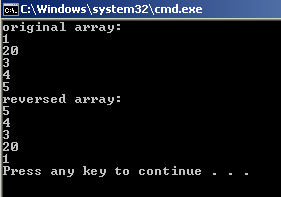
j = a[i];

a[i] = a[(N - 1) - i];

a[(N - 1) - i] = j;

}

}



**Exercise 4:**

Write a main program that reads the elements of a 2x2 integer matrix, **A**, from the user. It then performs the following tasks:

* prints the matrix row-by-row,
* adds matrix elements in row 0 and prints the sum, and
* adds matrix elements in column 0 and prints the sum.

For example, for matrix, **A**:



**The Sample Session will be as follows: *(Values in Red are entered by the user).***

Enter element at row 0 column 0: 1

Enter element at row 0 column 1: 3

Enter element at row 1 column 0: 7

Enter element at row 1 column 1: 9

The matrix entered by user is:

1 3

7 9

Sum of row 0 elements: 4

Sum of column 0 elements: 8

#include <iostream>

using namespace std;

void main()

{

int i,j,k,s;

k = 0;

s = 0;

int A[2][2];

for (i = 0; i < 2; i++)

{

for (j = 0; j < 2; j++)

{

cout << "Enter element at row " << i << " column " << j << ": ";

cin >> A[i][j];

}

}

cout << "The matrix entered by user is: " << endl;

for (i = 0; i < 2; i++)

{

for (j = 0; j < 2; j++)

{

cout << A[i][j] << " ";

}

cout << endl;

}

for (i = 0; i < 2; i++)

{

j = 0;

k = k + A[i][j];

}

for (j = 0; j < 2; j++)

{

i = 0;

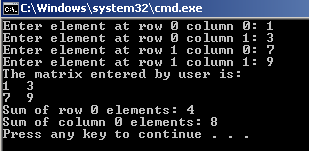
s = s + A[i][j];

}

cout << "Sum of row 0 elements: " << s << endl;

cout << "Sum of column 0 elements: " << k << endl;

}



**Exercise 5:**

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, print all those three variables. Moreover, print the address of variable **x**.

#include <iostream>

using namespace std;

void main()

{

int x, y, z;

int \*ptr;

x = 3;

y = 4;

z = 6;

ptr = &x;

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

cout << "Address of x is: " << ptr << endl;

ptr = &y;

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

ptr = &z;

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

}

